

tbvar: A Comprehensive Genome Variation Resource for *Mycobacterium tuberculosis*

A Major Project dissertation submitted

in partial fulfilment of the requirement for the degree of

Master of Technology

in

Bioinformatics

Submitted by

Heena Dhiman

(2K11/BIO/07)

Delhi Technological University, Delhi, India

Under the supervision of

Dr. Yasha Hasija



Department of Biotechnology
Delhi Technological University
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road,
Delhi-110042, INDIA



CERTIFICATE

This is to certify that the M. Tech. dissertation entitled “**tbvar: A comprehensive genome variation resource for *Mycobacterium tuberculosis***”, submitted by **HEENA DHIMAN (2K11/BIO/07)** in partial fulfilment of the requirement for the award of the degree of Master of Technology, Delhi Technological University (Formerly Delhi College of Engineering, University of Delhi), is an authentic record of the candidate’s own work carried out by her under my guidance.

The information and data enclosed in this thesis is original and has not been submitted elsewhere for honouring of any other degree.

Date:

Dr. Yasha Hasija

(Project Mentor)

Assistant Professor and Associate Head

Department of Biotechnology

Delhi Technological University

(Formerly Delhi College of Engineering, University of Delhi)

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to various people who have helped & supported me throughout this project. I would like to express my greatest appreciation to my guide and project mentor *Dr. Yasha Hasija*, for her exemplary guidance, monitoring and constant encouragement throughout the M. Tech. course. The assignments given by her from time to time not only made me capable of doing in depth analysis required to complete this project but has also given me a great experience that shall move along me in the journey of life on which I am about to embark. I am very grateful to her for giving me this opportunity to carry out such a novel and valuable research, providing me continuous support throughout this project and introducing me to my co-guide Dr. Vinod Scaria (Scientist – CSIR-IGIB).

I wish to express my sincere gratitude to *Dr. Vinod Scaria*, for the planning and development of this research work and allowing me to use the infrastructure of his lab. His willingness to give his time so generously has been very much appreciated.

I am particularly grateful for the assistance given by *Mr. Kandarp Joshi* (PhD Scholar – AcSIR). I am highly indebted to him for his guidance and constant supervision as well as being patient with my repetitive queries regarding this project.

My special thanks are extended to my classmates and the other PhD scholars at IGIB for providing useful critiques of this research work.

Lastly, I would like to thank my family members for their constant encouragement without which this assignment would not have been possible.

CONTENTS

TOPIC	Pg. No.
<i>LIST OF FIGURES</i>	<i>i</i>
<i>LIST OF TABLES</i>	<i>iii</i>
<i>LIST OF ABBREVIATIONS</i>	<i>iv</i>
1. ABSTRACT	1
2. INTRODUCTION	2
3. REVIEW OF LITERATURE	4
3.1 Tuberculosis	4
3.1.1 Latent TB Infection	4
3.1.2 TB Disease	4
3.1.3 Growth of tuberculosis	5
3.1.4 Molecular mechanisms of drug resistance	5
3.1.5 MTB Genomics	7
3.1.6 MTB Variomics	8
3.2 Next Generation Sequencing Technologies	10
3.2.1 Second generation HT-NGS platforms	12
3.2.2 Third generation HT-NGS platforms	14
3.2.3 Cost, Throughput, Accuracy and Completeness	16
3.2.4 Next Generation Sequencing – Promises and Challenges	18
3.3 Next Generation Sequencing Data and Analysis	19
3.3.1 Data Format	19
3.3.2 Data Assessment	20
3.3.3 Alignment Tools	21
3.3.4 Annotation tools	28
3.3.5 Existing Databases	36
4. METHODOLOGY	40
4.1 Datasets and Methods	40
4.2 Read Mapping and variant calling	41
4.3 Variant comparison	41
4.4 Variant annotation	42
4.5 Mapping Genes onto the Variants	42
4.6 Functional analyses of variations	43
4.7 Mapping of variants to regulatory regions	43
4.8 Mapping of variants to ncRNA	43
4.9 Mapping of variants to Sample information	44
4.10 Mapping of variants to information related to Drug Resistance	45

4.11 Mapping of variants to those available in existing databases	45
4.12 Database construction	46
4.13 Embedding JBrowse in the interface	50
5. RESULTS	51
5.1 Data Compilation	51
5.2 Database statistics	52
5.3 Genomic variations tend to saturation	54
5.4 Database features and navigation	55
5.4.1 Home Page	55
5.4.2 tbVar	56
5.4.3 Application of tbVAR: annoTB	61
5.4.4 Help Manual	65
5.4.5 Contact Page	66
6. DISCUSSION	67
7. CONCLUSION AND FUTURE PERSPECTIVE	69
8. REFERENCES	70
9. APPENDIX	76

LIST OF FIGURES

Fig. No.	Description	Page No.
1.	Molecular Mechanism of Drug Resistance	5
2.	Genome Map of <i>Mycobacterium tuberculosis</i>	8
3.	Automated Sanger Sequencing	10
4.	454 GS FLX Pyrosequencing	12
5.	Solexa GA Sequencing	12
6.	SOLiD Schema for Sequencing	12
7.	Three Leading Second Generation HT-NGS Platforms and Their Features	13
8.	Description of Fastq File Format	19
9.	General Mechanism for Deciphering Polymorphism	21
10.	Home Page of dbSNP	36
11.	Home Page of TBDB	37
12.	Search Page of MTCID	38
13.	Home Page of TBDReaMDB	39
14.	Downloading Mtb Gene Table from UCSC Table Browser	42
15.	Exporting ChIP-Seq Peaks Data from TBDB	43
16.	Downloading ncRNA Data from UCSC Table Browser	44
17.	Extracting Strain Information from Sequence Read Archive	44
18.	Extracting the Drug Resistance Data from TBDReamDB	45
19.	Constructing a Database on MySQL Workbench	46
20.	Developing a Table with Required Columns on MySQL Workbench	47
21.	Information Retrieval from Database by Web Browser Using CGI	48
22.	Summary of the Datasets and Methodology Used in Creating the Resource	49
23.	Comparison of the Variations in <i>M. tuberculosis</i> with respect to Other Variation Resources.	52
24.	Graphical Representation Showing Distribution of SNPs in Various Loci of the <i>M. tuberculosis</i> Genome	53
25.	Variations Plotted Across Subset of the Genomes.	54
26.	Home Page of the Web-Interface for tbvar	55
27.	Screenshot Showing Result Table and Information About Each Section of the Database	56
28.	Information Provided Under 'Genomic Variations' Tab	57
29.	Information Provided Under 'Gene Annotation' Tab	57
30.	Information Provided Under 'Functional Effects' Tab	58
31.	Information Provided Under 'Regulatory Variations' Tab	58
32.	Information Provided Under 'Strain Information' Tab	59
33.	Information Provided Under 'Drug Resistance' Tab	59

34.	Information Provided Under ‘ncRNA Loci’ Tab	60
35.	Information Provided Under ‘Genome Browser’ Tab	61
36.	Description of the annoTB Web-Page	62
37.	Information Provided Under annoTB Report Summary	62
38.	Information Related to ‘Drug Resistant Variations’ Provided Under annoTB Report	63
39.	Information Related to ‘Deleterious Variations’ Provided Under annoTB Report	63
40.	Information Related to ‘Syn/Non-Syn Variations’ Provided Under annoTB Report	63
41.	Information Related to ‘Regulatory Variations’ Provided under annoTB Report	64
42.	Information Related to ‘Novel Variations’ Provided Under annoTB Report	64
43.	Submission Form for Submitting Variant File that has been Loaded By the User in annoTB	64
44.	Screenshot of Web-Interface for the Reference Manual	65
45.	Screenshot of Web-Interface for the Contact Us Form	66

LIST OF TABLES

Table. No.	Description	Page No.
1.	Comparative Representation of the Specifications of the Next Generation Technologies	13
2.	Comparison Among the Various Applications of the Next Generation Sequencing Technologies	17
3.	Description of HTML Scripts Used to Develop the Web Interface of tbvar	48
4.	Description of Perl-CGI Scripts Used to Develop the Web Interface of tbvar	48
5.	Description of the Columns that have been Included While Compiling the Database	51

LIST OF ABBREVIATIONS

MTB	<i>Mycobacterium tuberculosis</i>
MTBC	<i>Mycobacterium tuberculosis</i> Complex
NGS	Next Generation Sequencing
SNP	Single Nucleotide Polymorphism
SIFT	Sorting Intolerant From Tolerant
MDR	Multi Drug Resistance
XDR	Extensive Drug Resistance
WHO	World Health Organization
INH	Isoniazid
RIF	Rifampin
PZA	Pyrazinamide
EMB	Ethambutol
Q	Phred Quality
MAQ	Mapping and Assembly with Qualities
BWT	Burrows–Wheeler Transform
BWA	Burrows–Wheeler Aligner
AAS	Amino Acid Substitution
SRA	Sequence Read Archive
ncRNA	Non-coding RNA

tbvar: A comprehensive genome variation resource for *Mycobacterium tuberculosis*

Heena Dhiman

Delhi Technological University, Delhi, India

1. ABSTRACT

Tuberculosis (TB) is the second highest cause of mortality after HIV/AIDS and is one of the leading public health problems in the developing world, caused by the fastidious pathogen *Mycobacterium tuberculosis* (MTB). The increasing resistance to anti-TB drugs and the recalcitrant nature of tenacious infections give rise to arduous challenges for the treatment of TB. Although, the advent of Next Generation Sequencing (NGS) has led to the discovery of thousands of Single Nucleotide Polymorphisms (SNPs) in clinical isolates of *Mycobacterium tuberculosis* complex (MTBC), this genetic variability amongst different isolates is poorly understood. MTBC strain variation is known to play a role in the outcome of TB infection and disease and can also affect the bacterial phenotype including drug resistance.

This work is aimed towards the analysis of high coverage resequencing datasets available in public domain. A data analysis pipeline was designed and used to reassemble, annotate and catalogue the SNPs in each of the datasets from over 400 different isolates. All the deciphered information was used to compile a comprehensive, well-curated and user-friendly database dedicated to the SNP data, along-with an interface for quick annotation of variations. Our analysis revealed a broad repertoire of more than 29,000 variations in MTB, in comparison to the H37Rv reference genome. 21,616 variations were found to be novel, significantly adding to the ensemble of known SNPs in MTB and 5,394 were predicted to be potentially deleterious in 2,407 genes as predicted by SIFT.

To the best of our knowledge tbvar is the largest and most comprehensive genome variation resource for *M. tuberculosis*. It not only offers a user friendly interface for annotating SNPs but also provides a starting point towards clinical application of variant information. The database is available as a free online resource at <http://genome.igib.res.in/tbvar/>

2. INTRODUCTION

Tuberculosis (TB), caused by the sole infectious agent *Mycobacterium tuberculosis* (MTB), is the second greatest killer worldwide next to HIV/AIDS. Fact Sheet, 2012 by World Health Organization reports that 8.7 million people became ill and 1.4 million died from TB in 2011. The standard 6 month course for TB treatment includes the prescription of four antimicrobial drugs –rifampin, pyrazinamide, isoniazid and ethambutol. However, the emergence of multi drug resistance (MDR) and extensive drug resistance (XDR) toward standard tuberculosis treatment has resulted in increasing severity of the disease.

The pronouncement of tuberculosis (TB) as a global public health emergency in 1993 (WHO, 2011) resulted in renewed efforts towards the analysis of the biology of the *Mycobacterium tuberculosis* complex (MTBC). Since long, the main research focus was on individual genes and proteins, but with the completion of the first *M. tuberculosis* genome sequence in 1998 (Cole *et. al.*, 1998) the doors for more comprehensive approaches opened up. In particular, comparative genomics studies have catered to develop a better insight into the genetic diversity in MTBC (Brosch *et. al.*, 2002; Mostowy *et. al.*, 2002; Comas *et. al.*, 2010) while Systems Biology tries to understand complex biological systems by integrating data from various disciplines (Breitling *et. al.*, 2010; Kirschner *et. al.*, 2010). There is increasing confirmation to the fact that, along-with human genetics and environmental factors, strain variation in MTBC also plays a role in the outcome of TB infection and disease (Coscolla and Gagneux, 2010). Hence, the need-of-the-hour is to better understand the global diversity of MTBC, and determine whether it has relevance for global TB control and if so, find out ways of doing it (Gagneux and Small, 2007; Comas and Gagneux, 2009).

The advent of next-generation DNA sequencing (NGS) methods is likely to facilitate this task by providing a new avenue to investigate pathogens in clinical settings. The recent years have seen sequencing of a large number of bacterial pathogens, including several strains of *M. tuberculosis* together with those sequenced in clinical settings (Wellcome Trust Sanger Institute, 2012). More than 3800 raw genome sequences of MTBC strains have already been deposited on public sequence read archives, and it is safe to assume that this number will continue to grow rapidly as sequencing costs keep decreasing (Stein, 2010; Wetterstrand, 2012). In contrast to the relative ease with which DNA sequencing data can be generated today, extracting useful information and compiling it in a user-friendly manner is less straightforward. Moreover, lack of a systematically curated resource for variations in the *M. tuberculosis* genome has significantly compromised the systematic comparison and interpretation of genomic variations in this organism. Clinical interpretation of the variations encoded by the genome has been one of the challenges, and necessitates systematic curation of genetic variations towards interpreting potential functional effects of these variations. Several TB-specific databases have been created over the past few years, including genome browsers, genotyping- and drug resistance databases, (Sharma and Surolia, 2011) but the necessity of a centralized and comprehensive repository for data on strain-specific genetic variation in MTBC and roadblocks towards systematically assembling and annotating

genomes on a common and comparable platform has been discussed in detail recently ([Stucki and Gagneux, 2013](#)).

In this report, we describe a comprehensive, well curated and user-friendly resource which stores systematically analyzed re-sequencing datasets of *M. tuberculosis* from various laboratories in public domain. This dataset encompasses over 29,000 variations from more than 450 strains that make it the most comprehensive compendium of genomic variations in *M. tuberculosis* as of now. We have been able to characterize potential genomic variations with functional consequences as well as their association with drug resistance using a systematic computational data analysis pipeline. The resource not only provides a near-comprehensive repertoire of common genomic variations in the organism but can also be potentially used for clinical applications. To the best of our knowledge tbvar is the largest and most comprehensive genome variation resources for *M. tuberculosis*. This resource is available for free access at <http://genome.igib.res.in/tbvar/>

3. REVIEW OF LITERATURE

3.1 Tuberculosis:

Tuberculosis, or TB, is an infectious bacterial disease caused by *Mycobacterium tuberculosis*, which most commonly affects the lungs. Infection with *M. tuberculosis* causes enormous worldwide morbidity and mortality. In 2011, there were an estimated 8.7 million new cases of TB (13% co-infected with HIV) and 1.4 million people died from TB, including almost one million deaths among HIV-negative individuals and 430 000 among people who were HIV-positive. TB is one of the top killers of women, with 300 000 deaths among HIV-negative women and 200 000 deaths among HIV-positive women in 2011. Geographically, the burden of TB is highest in Asia and Africa. India and China together account for almost 40% of the world's TB cases. About 60% of cases are in the South-East Asia and Western Pacific regions. The African Region has 24% of the world's cases and the highest rates of cases and deaths per capita. Worldwide, 3.7% of new cases and 20% of previously treated cases were estimated to have MDR-TB. There were an estimated 0.5 million cases and 64 000 deaths among children in 2011 (WHO-Fact Sheets, 2012).

Not everyone infected with *MTB* becomes sick. As a result, two TB-related conditions exist: latent TB infection and TB disease. Both latent TB infection and TB disease are preventable and treatable.

3.1.1 Latent TB Infection:

People with latent TB are infected with *Mycobacterium tuberculosis*, but they don't fall sick because the bacteria are not active. Latent TB infection has no symptoms, and they cannot spread the bacteria to others. However, if the bacteria become active in the body and multiply, the person will go from having latent TB infection to being sick with TB disease. For this reason, people with latent TB infection are often prescribed treatment to prevent them from developing TB disease. Four regimens are approved for the treatment of latent TB infection that includes:

- Isoniazid (INH)
- Rifampin (RIF)
- Rifapentine (RPT)

3.1.2 TB Disease:

The causative agent, *M. tuberculosis*, become active (multiplying in the body) if the immune system can't stop them from growing giving rise to [TB disease](#). This is when the disease becomes infectious and takes several drugs for 6 to 9 months for treatment. There are 10 drugs currently approved by the U.S. Food and Drug Administration (FDA) for treating TB. Of the approved drugs, the first-line anti-TB agents that form the core of treatment regimens include:

- Isoniazid (INH)
- Rifampin (RIF)
- Ethambutol (EMB)
- Pyrazinamide (PZA)

Regimens for treating TB disease have an initial phase of 2 months, followed by a choice of several options for the continuation phase of either 4 or 7 months (total of 6 to 9 months for treatment). It is very important that people who have TB disease finish the medicine, taking the drugs exactly as prescribed. If they stop taking the drugs too soon, they can become sick again; if they do not take the drugs correctly, the TB bacteria that are still alive may become resistant to those drugs. TB that is resistant to drugs is harder and more expensive to treat ([Kochi et. al., 1993](#)).

3.1.3 Growth of tuberculosis:

Among the factors that contribute to the continued growth of tuberculosis as a global health problem are the efficiency of human-to-human transmission by the aerosol route, the ability of the causal agent *M. tuberculosis* to persist and to progress despite development of host immune responses and the absence of a vaccine with reliable efficacy in preventing transmission of the infection. Moreover, although attempts to control tuberculosis through improved identification and treatment of infectious cases have been successful in some settings; similar approaches in other contexts have resulted in increasing rates of resistance to available anti-tuberculosis drugs.

3.1.4 Molecular mechanisms of drug resistance:

The emergence of Multi-Drug Resistant (MDR) and Extensively Drug Resistant (XDR) MTB has hampered the control of the disease. In order to control the drug resistance epidemic it is necessary to gain insight into how *M. tuberculosis* develops drug resistance. This knowledge will help us to understand how to prevent the occurrence of drug resistance as well as identifying genes associated with drug resistance of new drugs. The development of clinical drug resistance in TB is classified as acquired resistance when drug resistant mutants are selected as a result of ineffective treatment or as primary resistance when a patient is infected with a resistant strain ([Jarlier and Nikaido, 1994](#); [Blanchard, 1996](#)).

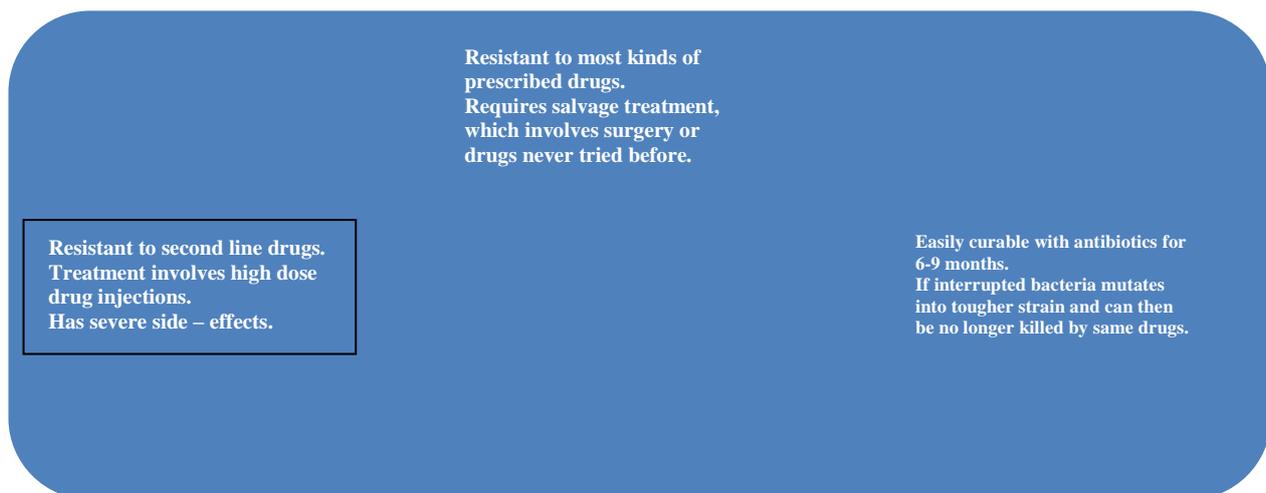


Fig. 1: Molecular mechanism of drug resistance

Mutations in the genome of *M. tuberculosis* that can confer resistance to anti-TB drugs occur spontaneously with an estimated frequency of 3.5×10^{-6} for INH and 3.1×10^{-8} for RIF.

Because the chromosomal loci responsible for resistance to various drugs are not linked, the risk of a double spontaneous mutation is extremely low: 9×10^{-14} for both INH and RIF. MDR TB is resistance of MTB to the first-line drugs, Rifampin and Isoniazid, while XDR TB is resistance to Isoniazid and Rifampin, Fluoroquinolone and at least one of three injectable second-line drugs (i.e., Streptomycin, Amikacin, Kanamycin, or Capreomycin) ([De Rossi et. al., 2006](#)).

First line drugs:

Any drug used in the anti-TB regimen is supposed to have an effective sterilizing activity that is capable of shortening the duration of treatment. Currently, a four-drug regimen is used consisting of INH, RIF, PZA and EMB. Resistance to first line anti-TB drugs has been linked to mutations in at least 10 genes; *katG*, *inhA*, *ahpC*, *kasA* and *ndh* for INH resistance; *rpoB* for RIF resistance, *embB* for EMB resistance, *pncA* for PZA resistance and *rpsL* and *rrs* for STR resistance.

Second line drugs used in TB treatment:

According to the WHO the following drugs can be classified as second line drugs: aminoglycosides (kanamycin and amikacin) polypeptides (capreomycin, viomycin and enviomycin), fluoroquinolones (ofloxacin, ciprofloxacin, and gatifloxacin), D-cycloserine and thionamides (ethionamide and prothionamide). Unfortunately, second-line drugs are inherently more toxic and less effective than first-line drugs.

Isoniazid, a prodrug, on activation interferes with the synthesis of essential mycolic acids by inhibiting NADH dependent enoyl-ACP reductase, which is encoded by *inhA*. Mutations in *katG* and *inhA*, or more often, in its promoter region, is considered to be the main cause for Isoniazid resistance ([Ramaswamy et. al., 2003](#)). Rifampin targets the β -subunit of RNA polymerase of MTB, where it binds and inhibits the elongation of messenger RNA. Mutations in the gene *rpoB*, that encode the β -subunit of RNA polymerase, are shown to be responsible for the resistance in clinical isolates of MTB ([Telenti et. al., 1993](#)). Pyrazinoic acid, the active moiety of Pyrazinamide, disrupts bacterial membrane energetics and inhibits membrane transport. Mutations in *pncA* are the main mechanisms for pyrazinamide resistance in MTB. Similarly, the genetic basis of resistance to Streptomycin, in MTB, is mostly due to mutations in *rrs* or *rpsL*, which produce alterations in the streptomycin binding site, which is more than 50% of the strains studied to date. The only target for Fluoroquinolone activity in MTB is Type II topoisomerase (DNA gyrase). Resistance to fluoroquinolones was the result of amino acid substitutions in the putative fluoroquinolone binding region in *gyrA* or *gyrB* ([Takiff et. al., 1994](#); [Silva et. al., 2003](#); [Aubry et. al., 2004](#)).

Disease caused by resistant bacteria fails to respond to conventional, first-line treatment. MDR-TB is treatable and curable by using second-line drugs, but second-line treatment options are limited and recommended medicines are not always available. The extensive chemotherapy required (up to two years of treatment) is more costly and can produce severe adverse drug reactions in patients. Around 650,000 cases of MDR-TB have been reported to

be present in the world in 2010, about 9% of which had XDR-TB. Annually, about 440 000 fell ill with MDR-TB and 150,000 die due to this form of tuberculosis ([Morris et. al., 2005](#)).

Therefore, new approaches to controlling tuberculosis are essential and would greatly benefit from an improved understanding of the biology of the bacteria and their interactions with their human hosts. In particular, understanding the factors that drive the evolution of *M. tuberculosis* and allow it to evade host defences may suggest unique opportunities to develop novel strategies against tuberculosis.

3.1.5 MTB Genomics:

MTB is a member of the *M. tuberculosis* Complex (MTBC), a closely related group of slow-growing pathogenic mycobacteria that includes *M. tuberculosis*, *Mycobacterium africanum*, *Mycobacterium bovis*, *Mycobacterium microti* and *Mycobacterium pinnipedii* (Comas et. al., 2010). The decade following the genome sequencing of *M. tuberculosis* genome has witnessed tremendous advances in the field of genomics. These changes have been propelled by significant improvements in scale, throughput and consequent drastic reduction in the cost of genome sequencing. Present genome sequencing technologies have provided a new avenue to investigate pathogens in clinical settings, which poses new challenges in comprehending the biology and the variations encoded by the pathogen.

Studies related to MTBC evolution have revealed that the *M. tuberculosis* genome appears to be a composite genome created by frequent horizontal gene transfer events in a broad, genetically diverse, progenitor species prior to an evolutionary bottleneck or selective sweep around 35,000 years ago ([Gillespie, 2002](#)). This recent clonal expansion with the concurrent absence of horizontal gene transfer explains the relatively high degree of genetic homogeneity (99.9%) observed between MTBC members despite differences in their phenotypic characteristics and host ranges. Whole genome sequencing of several *M. tuberculosis* strains has confirmed this genetic homogeneity and revealed many other interesting biological aspects. ([Fleischmann et. al., 2002](#); [Bentley, 2006](#); [Pellin et. al., 2012](#))

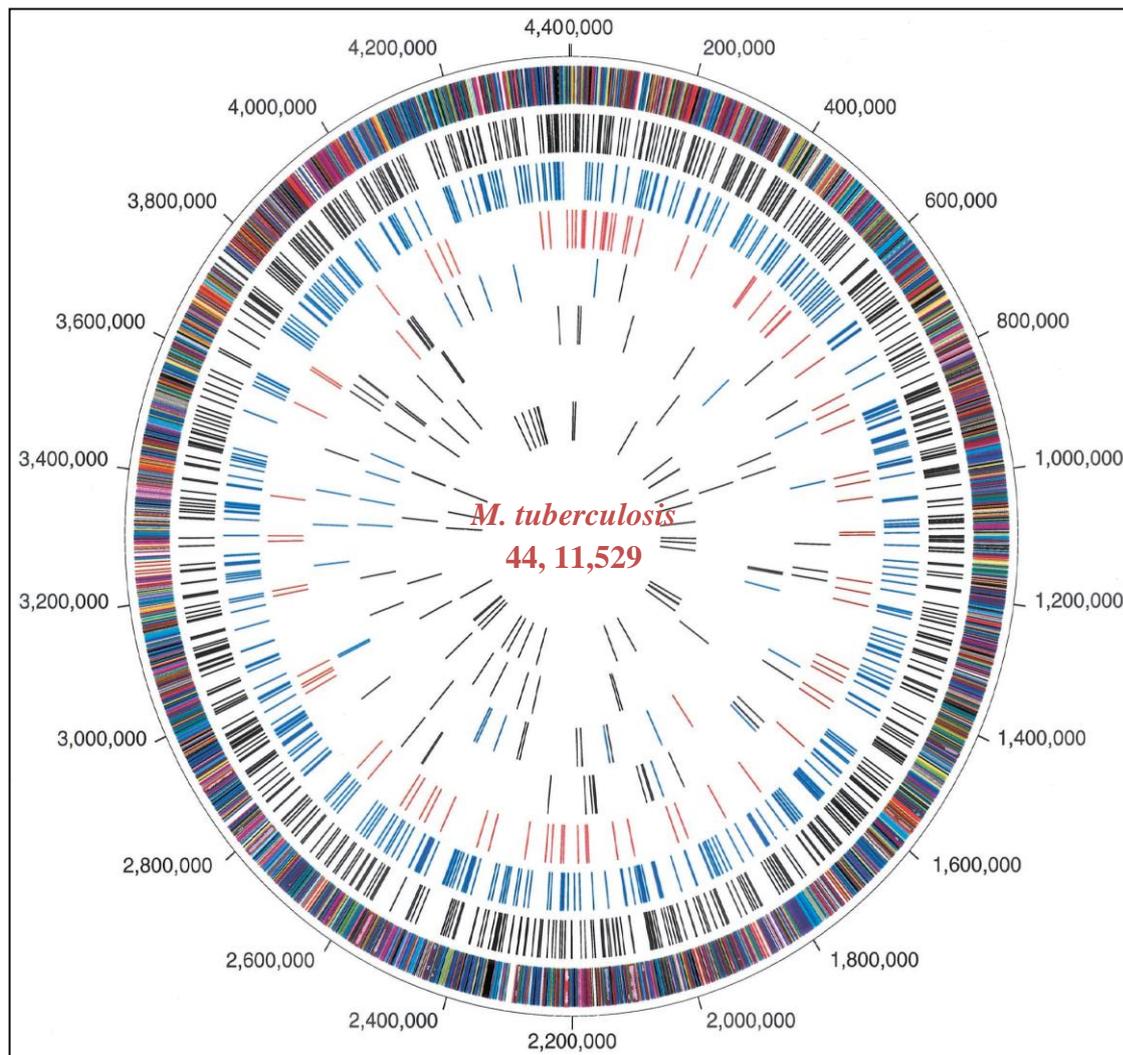


Fig. 2: Genome Map of *Mycobacterium tuberculosis* (Fleishland *et. al.*, 2002)

3.1.6 MTB Variomics:

Virulence and immunity are poorly understood in *Mycobacterium tuberculosis*. The genomic variability in *M. tuberculosis* has been majorly revealed through sequencing of multiple strains. A recent report (Ford *et. al.*, 2012) characterized the global diversity, circulating strain diversity and the evolution of *M. tuberculosis* through whole genome re-sequencing. Similarly the rate of mutations in active and latent infection of *M. tuberculosis* has been recently characterized by whole genome sequencing (Ford CB, 2011). The genome organization and evolution of the pathogen, especially in relation to its antigenic repertoire has also been characterized by sequencing multiple strains (Sasseti and Rubin, 2010). Furthermore, the variation between the H37Rv isolates maintained at multiple laboratories has also been characterized through a recent re-sequencing effort (Ioerger, Feng *et. al.* 2010). Genome sequencing of *M. tuberculosis* has also been recently applied extensively, unravelling the genome diversity of *M. tuberculosis* clinical isolates derived from a variety of geographical regions (Qi, Käser *et. al.*, 2009). In addition, genome sequencing of

paleontological samples has also been extensively used to trace genome evolution ([Qi, Käser et. al., 2009](#)) significantly adding to the spectrum of diversity information available from whole genome sequences of *M. tuberculosis*. Moreover, a number of strains associated to distinct phenotypes including drug resistance and mechanisms of evolution of drug resistance has also been extensively studied with respect to their genome sequence ([Niemann, Köser et. al., 2009](#)).

One of the surprises emerging from the analysis of the first sequenced *M. tuberculosis* genome (the laboratory strain H37Rv) was the discovery of two large gene families, designated *pe* and *ppe*, that in H37Rv comprise 99 and 69 members respectively and together account for around 10% of the organism's genomic coding potential. *pe* genes are characterised by the presence of a proline-glutamic acid (PE) motif at positions 8 and 9 within a highly conserved N-terminal domain consisting of around 110 amino acids. Similarly, *ppe* genes contain a proline-proline- glutamic acid (*ppe*) at positions 7–9 in a highly conserved N-terminal domain of approximately 180 amino acids. The C-terminal domains of both *pe* and *ppe* protein families are highly variable in both size and sequence and often contain repetitive DNA sequences that differ in copy number between genes ([McEvoy et. al., 2012](#)).

The presence of considerable sequence diversity in *M. tuberculosis* would provide a basis for comprehending pathogenesis, immune mechanisms, and bacterial evolution. Polymorphic genes are considered to be good candidates for virulence and immune determinants, since proteins that interact directly with the host are known to possess elevated divergence. Polymorphic sequences also serve as markers for phylogenetic and evolutionary studies. Such studies are currently limited by a paucity of known genetic markers ([Fleishmann et. al., 2002](#)).

SNPs carry functional information in addition to being valuable phylogenetic markers. The best-characterized “SNPs” in MTBC are drug resistance-conferring mutations. Drug resistance in MTBC is largely caused by single nucleotide mutations ([Musser, 1995](#); [Telenti, 1997](#); [Ramaswamy and Musser 1998](#); [Riska et. al., 2000](#)). In summary, thousands of SNPs are being identified in MTBC next-generation sequencing technologies. These SNPs may be a useful resource for phylogenetic and population genetic analyses and to study drug resistance.

3.2 Next Generation Sequencing Technologies

Genomic information has always been the focal point for genome-wide studies, but before the advent of Next Generation Sequencing (NGS) technology limitations in speed, resolution, scalability and throughput precluded researchers from the access to the immediate noesis of the genomic data. The commencement of NGS has not only reduced sequencing cost by orders of magnitude but has also catered to the other limitations, providing genome-scale sequence data with exquisite accuracy and resolution, thereby enabling to decode a number of human diseases. This could have been possible since the technology permitted the sequencing of whole genomes for obtaining global genomic information.

The year 1975 marks a landmark in biological sciences, especially clinical genomics, when Sanger and Coulson introduced a rapid technique for determining DNA sequences by primed synthesis with DNA polymerase. Since then several emerging technologies have turned up showing endeavor of providing solutions for fast and affordable genome sequencing. The modern DNA sequencing era began with the completion of the first human genome draft in June, 2000. In the following years, efforts were put in for improvising the rough draft in terms of coverage, number of gaps and the error rate, until the declaration of the essentially finished version of the human genome sequence by the International Human Genome Sequencing Consortium in April, 2003. From then on struggle has been going on to develop technologies capable of sequencing an entire human genome for \$1000. Several commercial ventures are racing towards this target with innovative highly developed strategies of “High-Throughput Next Generation Sequencing” (HT-NGS).

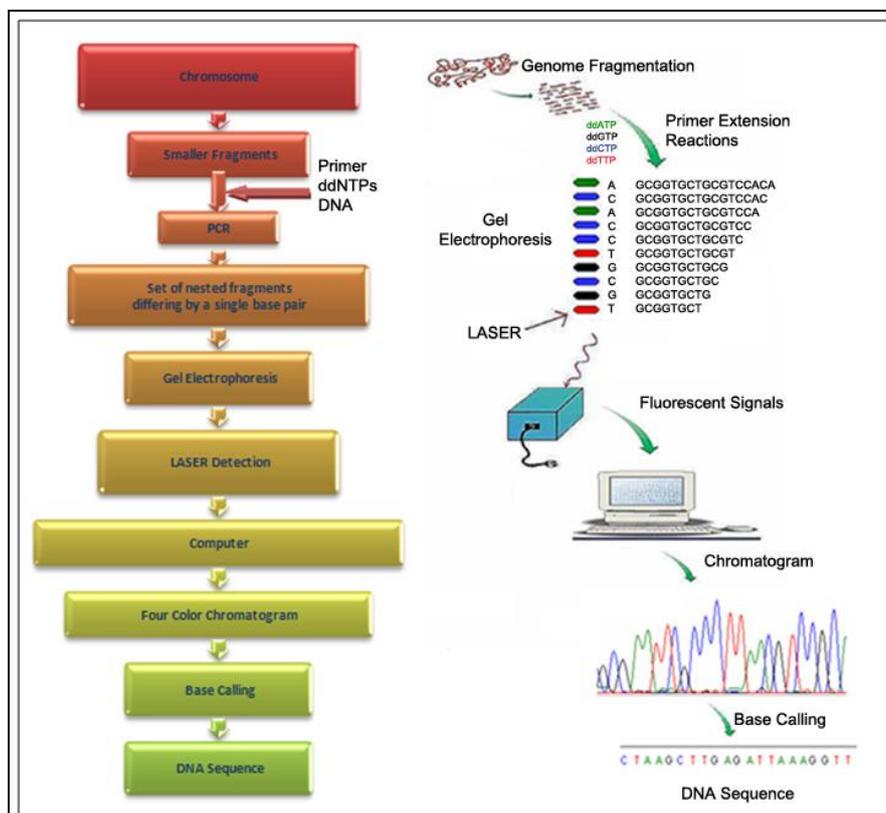


Figure 3: Automated Sanger Sequencing

The 'First Generation technology' employed Automated Sanger method while the newer techniques or the 'NGS Technology' makes use of different combination of strategies for template preparation, sequencing, imaging and genome alignment and assembly methods. Principle of HT-NGS involves sequencing DNA molecules in a flow-cell in massively parallel fashion in a step-wise iterative process or in a continuous real time manner ([Mardis, 2008](#); [Metzker, 2010](#)). In 2005, two new sequencing technologies were brought forward, both based on sequencing by synthesis, with an assurance to enhance traditional sequencing methods, the 454 system using pyrosequencing technology ([Margulies *et. al.*, 2005](#)), and the Solexa system, which detects fluorescence signals ([Porreca *et. al.*, 2007](#)).

HT-NGS platforms since 2005 has provided large numbers of low-cost reads thereby aiding in whole genome resequencing, RNA Sequencing, detection of genomic variants, genome-wide profiling of chromatin structure and epigenetic marks using methyl-seq, DNase-seq and ChIP-seq as well as Personal Genomics ([Pareek *et. al.*, 2011](#)). NGS can also be used to detect rare and unknown variants in genomic regions of interest in a cost-efficient way, and in a larger number of samples. With the advancement of modern bioinformatics tools and the ongoing progress of high throughput sequencing platforms at unparalleled swiftness, the target of sequencing individual genomes at a cost of \$1,000 each seems realistically viable in the near future.

The key steps of a sequencing project remain the same and include primarily preparation and amplification of template DNA, distribution of templates on a solid support, sequencing and imaging, base calling, quality control and data analysis. Sequencing depth and Breadth (Coverage) are the two major criterion and common measures for the amount of sequence data generated in a project. Sequencing depth, or coverage, is the average number of times each base in the genome is sequenced. Sequencing breadth, sometimes also referred to as genome coverage, is the percentage of the genome that is covered by sequence reads.

3.2.1 Second generation HT-NGS platforms

These include the three major NGS systems that are routinely used in many laboratories today:

1. Genome Sequencer from 454 Life Sciences (Launched in 2005).

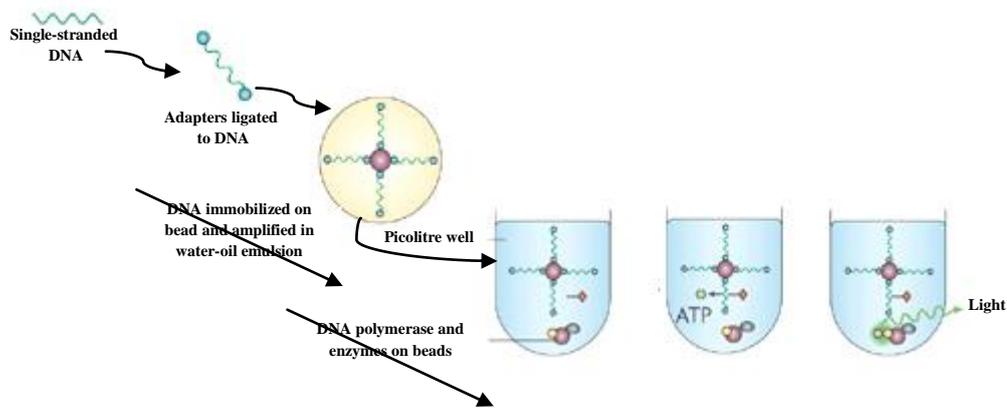


Fig. 4: 454 GS FLX pyrosequencing

2. Genome Analyzer, first conceived by Solexa and later further developed by Illumina (Launched in 2006).

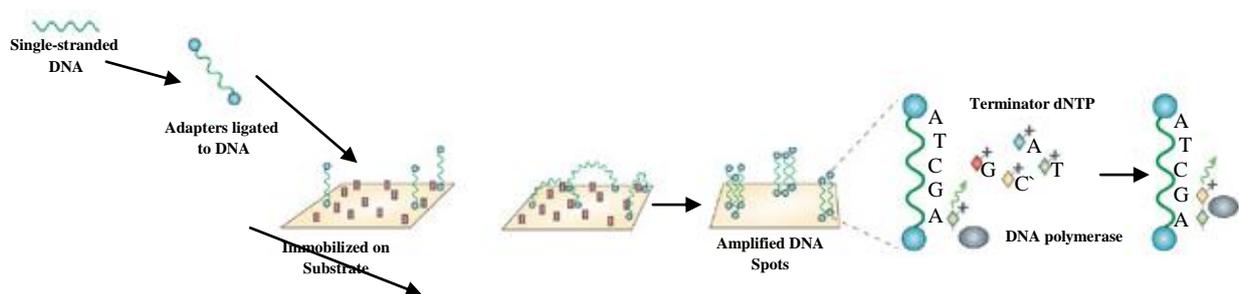


Fig. 5: Solexa GA Sequencing

3. SOLiD system from Applied Biosystems (Launched in 2007). ([Torres et. al., 2008](#))

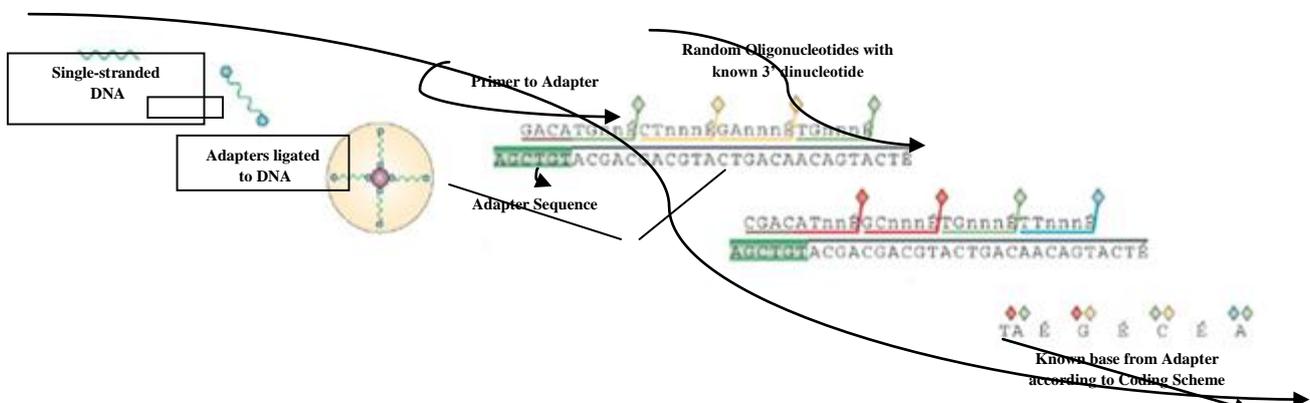


Fig. 6: SOLiD schema for sequencing

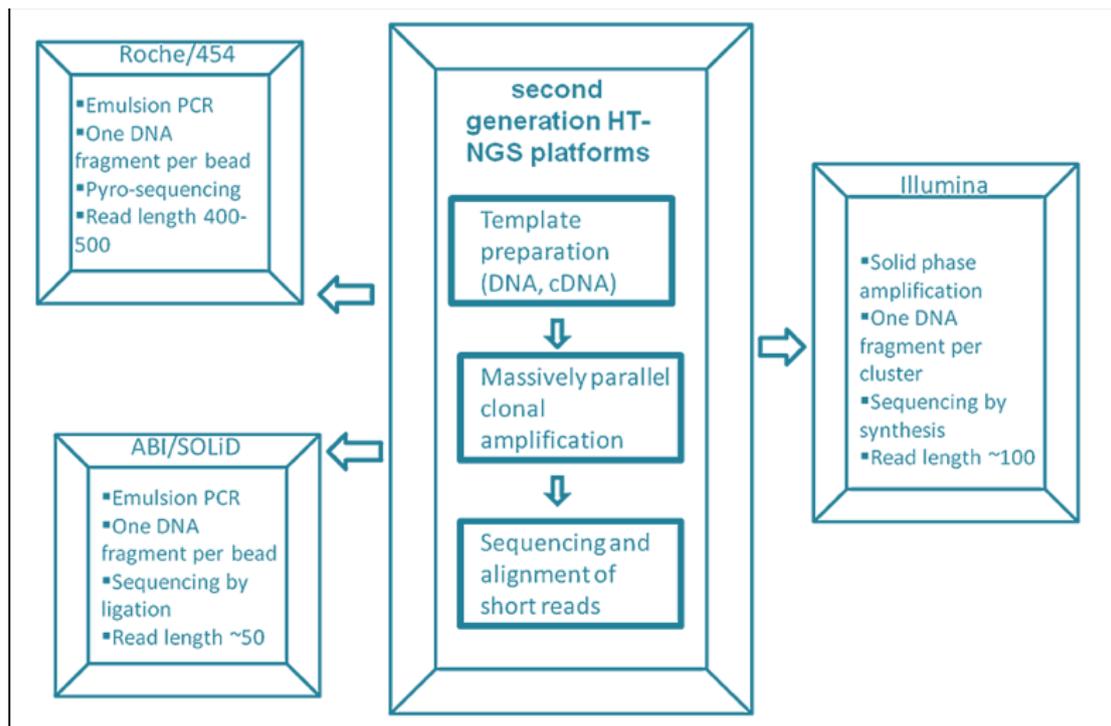


Fig. 7: Three leading second generation HT-NGS platforms and their features [Chandra et. al., 2011]

Roche can generate about five hundred million bases of raw sequencing data while Illumina and Solid can produce billions of bases in a single run. Their methodologies are based on parallel, cyclic interrogation of sequences from spatially separated clonal amplicons. Roche: pyrosequencing chemistry makes use of 26 μm oil-aqueous emulsion bead, SOLiD: sequencing by sequential ligation of oligonucleotide probes uses 1 μm clonal bead and Illumina: sequencing by reversible dye terminators uses clonal bridge for carrying out their technology (Pareek et. al., 2011). So, the underlying principle is amplification of DNA fragments using emulsion PCR, to make the light signal strong enough for reliable base detection by the CCD cameras.

Table I. Comparative representation of the specifications of the next generation technologies (Black M. and Print C., 2010)

Technology	Reads/run	Average read length	Estimated Time per run	Data output per run
Roche GS-FLX (454)	1.3 million	400 bp	10 hours	500 MB
Roche GS-Junior	10,000	400 bp	10 hours	35 MB(filtered)
Illumina 1G (solexa)	250 million	100 bp X 2	5 days	25 GB
Illumina HiSeq 2000	1 billion	100 bp X 2	8 days	200 GB
SOLiD (ABI)	1.4 million	50 bp X 2	4-6 days	100 GB

3.2.2 Third generation HT-NGS platforms

These platforms are based on sequencing from a single DNA molecule. Since PCR amplification may introduce base sequence errors, favor certain sequences over others, thereby changing the abundance and relative frequency of various DNA fragments that were present before amplification. Thus, it is required that the sequence should be determined directly from a single DNA molecule, without the need for PCR amplification and its ability for alteration of abundance levels.

3.2.2.1 Heliscope™ single molecule sequencer

It was the first commercial single molecule sequencing system licensed by Helicos Biosciences in 2007. It is based on the true Single Molecule Sequencing (tSMS) technology that begins with DNA library preparation through DNA shearing and addition of poly-(A) tail to fragmented DNA. This is followed by hybridization of DNA fragments to poly-(T) oligonucleotides and sequencing in parallel. It is capable of sequencing 28 Gb in a single sequencing run with read length of 55 bases in about 8 days ([Ozsolak et. al., 2010](#)).

3.2.2.2 Single molecule real time (SMRT™) sequencer

It is designed by the Pacific Biosciences, based on single molecule real time sequencing by synthesis method provided on the sequencing chip containing thousands of zero-mode waveguides (ZMWs). During the sequencing reaction, the DNA fragment is elongated by a single DNA polymerase, which is attached to the bottom of each ZMW, with dNTP's that are fluorescently labeled at the terminal phosphate moiety. CCD array is used for determining the DNA sequence based on fluorescence nucleotide detection. SMRT analyzer is capable of obtaining 100 Gb in an hour with reads longer than 1000bp in a single run.

3.2.2.3 RNAP Sequencer

This is another single molecule DNA sequencing approach, wherein RNA Polymerase (RNAP) is attached to a polystyrene bead and the distal end of the DNA fragment is attached to another bead. When RNAP interacts with the DNA fragment inside an optical trap, the length of DNA between the two beads gets altered. This causes displacement of the beads that is registered by the instrument. The sequence information can be deduced by aligning four displacement records produced in the same way like primers used in Sanger Sequencing. Calibration is done using the known sequences flanking to the unknown sequenced fragment ([Greenleaf and Block, 2006](#)).

3.2.2.4 Nanopore DNA Sequencer

Unlike other methods DNA sequencing here is free from nucleotide labelling and detection. DNA translocation studies from number of different artificial nanopores, forms the basis of this method. Modulation of the ionic current as DNA molecule traverses the pore reveals the

characteristics and parameters like length, diameter and conformation of the molecule. The time period for which the current is blocked by the nucleotide is characteristic for each base and enables the DNA sequence to be determined ([Astier et. al., 2006](#); [Rusk, 2009](#)).

3.2.2.5 Real Time Single Molecule DNA Sequencer

Specially engineered DNA Polymerase by the VisiGen Biotechnologies acts as a real-time sensor for nucleotides modified with a donor fluorescent dye. Each of the four nucleotides to be integrated is modified with different acceptor dye. During synthesis correct nucleotide found enters the active site of the enzyme making the donor dye come in close contact with the acceptor dye on the nucleotides, thus transferring energy from donor to acceptor dye giving rise to FRET signal. Base sequences are determined according to the variation in signal frequency. This technology could generate around 4GB of data per day.

2.2.2.6 Multiplex Polony technology

This technology is run by Personal Genome Project under the lead of Prof. G Church's research group. It employs parallel sequencing of hundreds of sequencing templates deposited onto thin agarose layers. It is capable of generating 10-35Gbp per module in a run of 2.5 days. This is achievable at a 10-fold lower cost with large reduction of reaction volumes and lesser amount of reagents ([Mitra et. al., 2003](#); [Shendure et. al., 2005](#)).

3.2.2.7 Ion Torrent Sequencing Technology

The technology makes use of chemical and digital information collectively, thus enabling faster, simpler and massively scalable sequencing. Release of hydrogen ion as a by-product during incorporation of a nucleotide into a DNA strand by a polymerase, forms the basis of this technology. Different DNA templates are kept in different micro machined wells, below which is an ion sensitive layer. Charge from the released hydrogen ion changes the pH of the solution that can be detected directly by the ion sensor. It allows multiplexing amplicons, sequencing transcriptome, small RNA, CHIP-Seq paired end reads and methylation reads ([Rothberg et. al., 2011](#)).

3.2.3 Cost, Throughput, Accuracy and Completeness

Cost, throughput, accuracy and completeness are interrelated, and efforts to decrease project cost or to increase throughput have sometimes been accompanied by a reduction in accuracy or completeness. The challenge for new technologies is to achieve massive improvement in one or more of these components without compromising the others ([Bentley, 2006](#)).

Cost

The single most effective constraint that modifies cost is massive parallelisation. Sequencing by synthesis on arrays has already achieved a parallelization of 10⁵–10⁷ reactions, in contrast to capillary systems (96 or 384 channel). Such methods can support in excess of 10⁸ reactions per experiment, although with shorter read lengths and hence fewer bases of sequence per reaction.

To achieve high data-density miniaturisation is a significant contributor to cost reduction, assuming comparable rates of throughput between systems. The potential increase in data density has a positive impact on reagent cost, while reagent volumes are significantly reduced, whereas it may be assumed that the instrument costs and concentrations of reagents in polymerase reactions are generally comparable, with less than tenfold variance between platforms.

Throughput

The intrinsic throughput of a sequencing system depends on speed of detection and degree of parallelisation. A capillary sequence reads 0.17 bases per second per channel, thus for 96 channels the total throughput in continuous operation comes out to be 17 bases per second, limited by the rate of electrophoresis only. Sequencing by synthesis on arrays has a much slower cycle time since reaction chemistry is carried out in-between read-outs at each cycle. This time penalty is compensated for by the degree of parallelisation, making the total throughput in continuous operation of current systems between 1400 and 4000 bases per second.

Accuracy

Accuracy of raw sequence data generated from capillary sequencing over most of the length of each read is measured using Phred algorithm ([Ewing *et. al.*, 1998](#)). The numerical score or the quality value provides an estimate of the error probability for each base-call in a raw read ([Ewing and Green, 1998](#)).

Multiple reads from the same sample can be aligned and used to obtain a consensus base-call at each position. Consensus depth provides high confidence base-calling, and the quality score of 15 may be considered sufficient (International Human Genome Sequencing Consortium, 2004). Aligned reads, and consensus and individual base-quality values can also

be used for calling SNP alleles. By contrast, if single reads are used to call a SNP with high confidence, a much higher threshold is applied in selecting the raw data that is used to support the base-call (The International SNP Map Working Group, 2001).

Similarly, consensus scoring and quality values can be used to call SNPs from base-by base sequencing and pyrosequencing data as well which can further be used to derive quality metrics. For all methods, it is necessary to calibrate quality values empirically, by generating large amounts of sequence data from a known template, measuring the frequency of incorrect base-calls and assessing the validity of various quantitative parameters in the raw data in correlation with observed error rates. More work is required in this area to assess the accuracy of each sequencing method, to look for loss of accuracy in particular sequence contexts for each method, and to enable assembly of sequence data obtained using more than one technology.

Completeness

Long reads of high accuracy provide a very high level of completeness in most sequencing projects. The completeness of the reference human genome (The International SNP Map Working Group, 2001) demonstrates the utility of long reads generated by Sanger sequencing. The availability of a reference genome sequence renders short reads very powerful as a means to obtain re-sequencing data. A short read is required to be long enough and sufficiently accurate, so as to align to the correct position in the reference uniquely. From simulations ([Whiteford et. al., 2005](#)), reads of length 25–30 bases can be aligned uniquely to cover 80% of the human genome sequence. For 1 Mb human genomes or 4 Mb bacterial genomes, reads can be aligned uniquely to 95–99% of all positions in the sequence. 25–30 base reads can potentially be used for de novo assembly. The use of short reads is not supposed to compromise the degree of completeness that can be obtained so greatly. The concept of a random shotgun phase proved to be very successful in producing ‘finished’ sequence of much better quality and utility than ‘unfinished’ or ‘draft’ sequence (International Human Genome Sequencing Consortium, 2004).

Table II. Comparison among the various applications of the Next generation sequencing technologies

Sequencers	454	Illumina	SOLiD
Resequencing		Yes	Yes
De novo	Yes	Yes	
Cancer	Yes	Yes	Yes
Array	Yes	Yes	Yes
High GC sample	Yes	Yes	Yes
Bacterial	Yes	Yes	Yes
Large genome	Yes	Yes	
Mutation Detection	Yes	Yes	Yes

3.2.4 Next Generation Sequencing – Promises and Challenges

In spite of the advantages next generation sequencing offers, there are a few limitations to this technology:

1. **Shorter read lengths compared to the Sanger method** – This is a major drawback of this technology over Sanger sequencing. De novo assembly of genome is difficult; hence this technology better serves as a genome “re-sequencing” tool.
2. **Repetitive DNA** – Almost 50% of the human genome has repetitive DNA. Owing to shorter read lengths, ambiguities in alignment and assembly arise in the areas of repeats.
3. **Data crunch** – Large volumes of data are generated and analyses is time consuming and expensive. Data analyses may represent the rate-limiting step in next generation sequencing. “Anybody can go out and buy an instrument. And the protocols of sequencing are fairly well worked out, although it requires a lot of training. However, it’s the handling of that information, of analyzing and distilling it, that’s still very challenging,” said Metzker.
4. **Non-uniform representation of genomic regions** - Random generation of sequencing reads because of intervening and unavoidable factors such as platform biases, sample handling, variations per run, etc makes the genomic data highly non-uniform. Thus, a certain amount and type of input data, along with a specified reference sequence is usually required in order to comprehend what proportion of the whole genome can be correctly ascertained ([Ajay et. al., 2011](#)).
5. **Difference in accuracy and precision** - Genome Sequencing techniques shows difference in accuracy and precision depending on the variations in sequencing chemistry, coverage (read-length and insert size), alignment and variant-calling algorithms of different platforms. Since each platform has its own strength and weakness, data from different platforms when merged, improves the overall accuracy. It has been shown that CG is less uniform in coverage than Illumina ([Metzker, 2010](#); [Kim et. al., 2013](#)).
6. **Cancer research** - Sequencing accuracy may depend on the quality of DNA from Formalin Fixed Paraffin Embedded (FFPE) tissues, which can be highly variable. Moreover the ability of NGS to distinguish sequence artifacts from low frequency mutations from FFPE samples deserves further validation (Debora and Christos, 2013).
7. **Mapping Studies and Comparative Genomics** - Mapping studies and comparative genome assembly require anchoring, as one of the important steps in comparison of assembled genomes that are evolutionarily related. However, the sheer numbers of fragments being produced through sequencing as well as the increasing size of the reference sequence have resulted in a computational bottleneck, necessitating the development of techniques to resolve this issue ([Ajay et. al., 2011](#)).

Despite these limitations, the next generation sequencing has revolutionized the fields of research and medicine. It has made possible large-scale studies such as the Encyclopedia of DNA Elements (ENCODE) project that aims to decipher functional elements encoded in the human genome.

3.3 Next Generation Sequencing Data and Analysis

In the coming years, all previous genotyping methods for MTBC are expected to get at least partially replaced by whole genome sequencing. Whole-genome resequencing is an essential research tool to characterize genetic variation in context of complex disease, pathogenicity, evolution and individuality.

3.3.1 Data Format

The FASTQ format allows the storage of both sequence and quality information for the sequencing reads. This is a compact text-based format that has become the de facto standard for storing data from next generation sequencing experiments, especially for Sanger Institute. Although Solexa/Illumina read file looks quiet alike, the only difference lies in scaling of the qualities. In the quality string, presence of any character with ASCII code higher than 90, depicts Solexa/Illumina format.

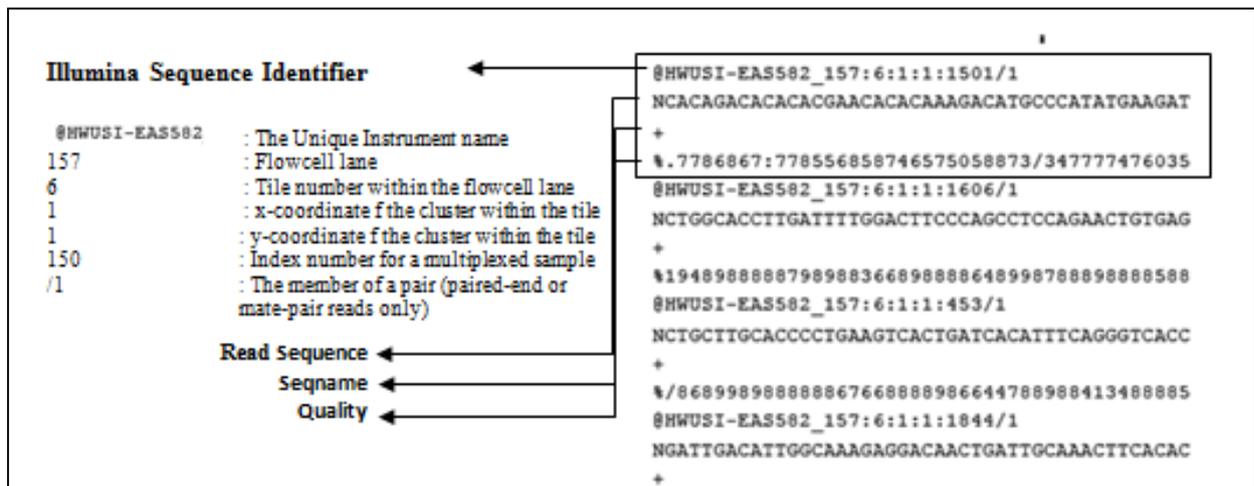


Fig.8: Description of fastq file format

The “Seqname” following '+' is optional, but if it appears right after '+', it should be identical to the sequence following '@'.

The length of “Read sequence” is identical to the length of “Quality”.

Each character in “Quality” represents the Phred quality of the corresponding nucleotide in “Read sequence”.

If the Phred quality is \$Q, which is a non-negative integer, the corresponding quality character can be calculated with the following Perl code:

$$\$q = \text{chr}(\$Q \leq 93 ? \$Q : 93) + 33;$$

where **chr()** is the Perl function to convert an integer to a character based on the ASCII table.

Conversely, given a character \$q, the corresponding Phred quality can be calculated with:

$$\$Q = \text{ord}(\$q) - 33;$$

where **ord()** gives the ASCII code of a character.

The syntax of Solexa/Illumina read format is almost identical to the FASTQ format, but the qualities are scaled differently. Given a character \$sq, the following Perl code gives the Phred quality \$Q:

$$\$Q = 10 * \log (1 + 10 ** (\text{ord}(\$sq) - 64) / 10.0)) / \log(10);$$

3.3.2 Data Assessment

Accuracy of raw sequence data generated from capillary sequencing over most of the length of each read is measured using Phred algorithm ([Ewing and Green, 1998](#)). It indicates the error probability for calling a given base by the sequencer. Historically this algorithm was used to determine Sanger sequencing accuracy. It indicates the error probability for calling a given base by the sequencer. While NGS metrics varies from those of Sanger Sequencing, the Phred Quality scoring scheme still remains the same. Parameters relevant to a particular sequencing chemistry are analyzed for a large training data set of known accuracy. The resulting quality score lookup tables are then made in use to assess quality score for de novo NGS data in Real time on Illumina platforms.

Quality scores or Q scores are logarithmically related to base calling error probabilities (P):

$$Q = -10 \log_{10} P$$

Table II: Quality scores and Base Calling Accuracy

Phred Quality Score	Probability of Incorrect Base Calling	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%
50	1 in 100000	99.999%

A Q score of 30 for a base is equivalent to the probability of an incorrect base being called 1 in 1000 times, for which base call accuracy is 99.9%.

A lower base call accuracy of 99% increases the incorrect base call probability to 1 in 100, which means every 100bp sequencing read will likely contain an error.

So, when sequencing quality reaches Q30 almost the entire read is supposed to be perfect, with no errors and ambiguities. Thus, Q30 is considered to be a benchmark to assess quality in NGS.

Low Q scores can increase false positive variant calls that can cause inaccurate conclusions and higher costs for validation experiments. The level of accuracy is ideal for a range of sequencing applications, including clinical research.

3.3.3 Alignment Tools

With the advent of Next Generation Sequencing technologies, many programs have come up in the last few years for aligning short sequencing reads to a reference genome. Most of them are applicable for short reads (100bp) but inefficient for reads greater than 200bp since the algorithms are specifically tuned for short queries with low sequencing error rate ([Li and Durbin, 2010](#)). The extensive amount of short reads being generated from the new DNA sequencing technologies call for development of fast and accurate read alignment programs.

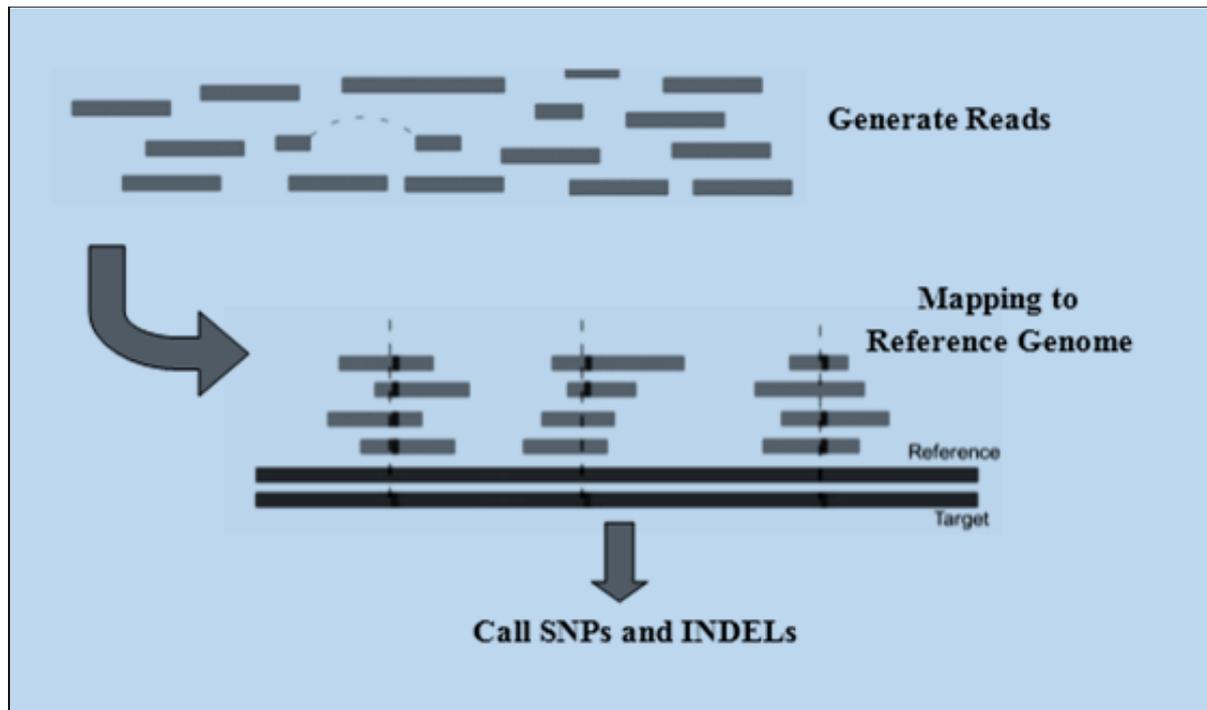


Fig. 9: General mechanism for deciphering polymorphism

MAQ is one of the first generation hash table based methods for accurate, feature rich and fast alignment of short reads from a single individual. BWA is another alignment package that is based on backward search with Burrows–Wheeler Transform (BWT), allowing mismatches and gaps. Bowtie is yet another ultrafast, memory-efficient short read aligner geared toward quickly aligning large sets of short reads to large genomes, by indexing the genome with a Burrows-Wheeler index to keep its memory footprint small.

MAQ - Mapping and Assembly with Qualities

MAQ can build assemblies by mapping shotgun short reads to a reference genome, using quality scores to derive genotype calls of the consensus sequence. It makes full use of mate-pair information and estimates the error probability of each read alignment. The program rapidly aligns short reads to the reference genome, call consensus sequences including SNP and indel variants, simulate diploid genomes and read sequences, and post-process the results in various ways. Error probabilities are also derived for the final genotype calls, using a Bayesian statistical model that incorporates the mapping qualities, error probabilities from the raw sequence quality scores, sampling of the two haplotypes, and an empirical model for

correlated errors at a site. Both read mapping and genotype calling are evaluated on simulated data and real data.

Alignment stage: MAQ first searches for the ungapped match with lowest mismatch score, defined as the sum of qualities at mismatching bases. To speed up the alignment, MAQ only considers positions that have two or fewer mismatches in the first 28 bp (default parameters). Sequences, that fail to reach a mismatch score threshold but whose mate pair is mapped, are searched with a gapped alignment algorithm in the regions defined by the mate pair. To evaluate the reliability of alignments, MAQ assigns each individual alignment a phred-scaled quality score (capped at 99), which measures the probability that the true alignment is not the one found by MAQ. MAQ always reports a single alignment, and if a read can be aligned equally well to multiple positions, MAQ will randomly pick one position and give it a mapping quality zero. Because their mapping score is set to zero, reads that are mapped equally well to multiple positions will not contribute to variant calling. However, they do give information on copy number of repetitive sequences and on the fraction of reads that can be aligned to the genome, and can easily be filtered out for downstream analysis if desired. Mapping quality scores and mapping all reads that match the genome even if repetitive are where MAQ differs from most other alignment programs. MAQ fully utilizes the mate-pair information of paired reads. It is able to use this information to correct wrong alignments, to add confidence to correct alignments, and to accurately map a read to repetitive sequences if its mate is confidently aligned. With paired-end reads, MAQ also finds short insertions/deletions (indels) from the gapped alignment described above.

SNP calling stage: MAQ produces a consensus genotype sequence from the alignment. The consensus sequence is inferred from a Bayesian statistical model, and each consensus genotype is associated with a phred quality that measures the probability that the consensus genotype is incorrect. Potential SNPs are detected by comparing the consensus sequence to the reference and can be further filtered by a set of predefined rules. These rules are designed to achieve the best performance on deep human resequencing data and aim to compensate for simplifications and assumptions used in the statistical model (e.g., treating neighbor positions independently).

Basic Commands:

fasta2bfa: Convert sequences in FASTA format to Maq's binary FASTA (BFA) format.

```
maq fasta2bfa in.ref.fasta out.ref.bfa
```

fastq2bfq: Convert reads in FASTQ format to Maq's BFQ (binary FASTQ) format.

```
maq fastq2bfq [-n nreads] in.read.fastq out.read.bfq|out.prefix
```

OPTIONS

```
-n INT    number of reads per file [not specified]
```

map: Map reads to the reference sequences.

```
maq map [-n nmis] [-a maxins] [-c] [-1 len1] [-2 len2] [-d adap3] [-m mutrate] [-u unmapped] [-e maxerr] [-M c/g] [-N] [-H allhits] [-C maxhits] out.aln.mapin.ref.bfa in.read1.bfq [in.read2.bfq] 2> out.map.log
```

OPTIONS

- n INT Number of maximum mismatches that can always be found [2]
- a INT Maximum outer distance for a correct read pair [250]
- A INT Maximum outer distance of two RF paired read (0 for disable) [0]
- c Map reads in the color space (for SOLiD only)
- 1 INT Read length for the first read, 0 for auto [0]
- 2 INT Read length for the second read, 0 for auto [0]
- Mutation rate between the reference sequences and the reads [0.001]
- m FLOAT
- d FILE Specify a file containing a single line of the 3'-adapter sequence [null]
- u FILE Dump unmapped reads and reads containing more than nmis mismatches to a separate file [null]
- e INT Threshold on the sum of mismatching base qualities [70]
- H FILE Dump multiple/all 01-mismatch hits to FILE [null]
- C INT Maximum number of hits to output. Unlimited if larger than 512. [250]
- M c|g methylation alignment mode. All C (or G) on the forward strand will be changed to T (or A). This option is for testing only.
- N store the mismatch position in the output file out.aln.map. When this option is in use, the maximum allowed read length is 55bp.

mapmerge: Merge a batch of read alignments together.

maq mapmerge out.aln.map in.aln1.map in.aln2.map [...]

assemble: Call the consensus sequences from read mapping.

OPTIONS

- t FLOAT Error dependency coefficient [0.93]
- r FLOAT Fraction of heterozygotes among all sites [0.001]
- s Take single end mapping quality as the final mapping quality; otherwise paired end mapping quality will be used
- p Discard paired end reads that are not mapped in correct pairs
- m INT Maximum number of mismatches allowed for a read to be used in consensus calling [7]
- Q INT Maximum allowed sum of quality values of mismatched bases [60]
- q INT Minimum mapping quality allowed for a read to be used in consensus calling [0]
- N INT Number of haplotypes in the pool (≥ 2) [2]

indelpe: Call consistent indels from paired end reads.

maq indelpe in.ref.bfa in.aln.map > out.indelpe

The output is TAB delimited with each line consisting of:

- Chromosome
- Start position
- Type of the indel
- Number of reads across the indel
- Size of the indel and inserted/deleted nucleotides (separated by colon)
- Number of indels on the reverse strand
- Number of indels on the forward strand
- 5' sequence ahead of the indel
- 3' sequence following the indel
- Number of reads aligned without indels

At the 3rd column, type of the indel:

- * Indicates the indel is confirmed by reads from both strands
- + Means the indel is hit by at least two reads but from the same strand
- Shows the indel is only found on one read
- . Means the indel is too close to another indel and is filtered out.

Commands for extracting information:

mapview: Display the read alignment in plain text.

```
maq mapview [-bN] in.aln.map > out.aln.txt
```

OPTIONS

- b Do not display the read sequence and the quality
- N To display the positions where mismatches occur. This flag only works with a .map file generated by 'maq map -N'.

For reads aligned before the Smith-Waterman alignment, each line consists of read name, chromosome, position, strand, insert size from the outer coordinates of a pair, paired flag, mapping quality, single-end mapping quality, alternative mapping quality, number of mismatches of the best hit, sum of qualities of mismatched bases of the best hit, number of 0-mismatch hits of the first 24bp, number of 1-mismatch hits of the first 24bp on the reference, length of the read, read sequence and its quality. Alternative mapping quality always equals to mapping quality if the reads are not paired. If reads are paired, it equals to the smaller mapping quality of the two ends. This alternative mapping quality is actually the mapping quality of an abnormal pair.

The fifth column, paired flag, is a bitwise flag. Its lower 4 bits give the orientation: 1 stands for FF, 2 for FR, 4 for RF, and 8 for RR, where FR means that the read with smaller coordinate is on the forward strand, and its mate is on the reverse strand. Only FR is allowed for a correct pair. The higher bits of this flag give further information. If the pair meets the paired end requirement, 16 will be set. If the two reads are mapped to different chromosomes, 32 will be set. If one of the two reads cannot be mapped at all, 64 will be set. The flag for a correct pair always equals to 18.

For reads aligned by the Smith-Waterman alignment afterwards, the flag is always 130. A line consists of read name, chromosome, position, strand, insert size, flag (always 130), position of the indel on the read (0 if no indel), length of the indels (positive for insertions and negative for deletions), mapping quality of its mate, number of mismatches of the best

hit, sum of qualities of mismatched bases of the best hit, two zeros, length of the read, read sequence and its quality. The mate of a 130-flagged read always gets a flag 18. Flag 192 indicates that the read is not mapped but its mate is mapped. For such a read pair, one read has flag 64 and the other has 192.

mapcheck: Read quality check.

```
maq mapcheck [-s] [-m maxmis] [-q minQ] in.ref.bfa in.aln.map > out.mapcheck
```

- s Take single end mapping quality as the final mapping quality
- m INT Maximum number of mismatches allowed for a read to be counted [4]
- q INT Minimum mapping quality allowed for a read to be counted [30]

The mapcheck first reports the composition and the depth of the reference. The first column indicates the position on a read. Following four columns which show the nucleotide composition, substitution rates between the reference and reads will be given. These rates and the numbers in the following columns are scaled to 999 and rounded to nearest integer. The next group of columns shows the distribution of base qualities along the reads at a quality interval of 10. Decay in quality can usually be observed, which means bases at the end of read are less accurate. The last group of columns presents the fraction of substitutions for read bases at a quality interval. This measures the accuracy of base quality estimation.

pileup: Display the alignment in a 'pileup' text format.

```
maq pileup [-spvP] [-m maxmis] [-Q maxerr] [-q minQ] [-l sitefile] in.ref.bfa in.  
aln.map > out.pileup
```

OPTIONS

- s Take single end mapping quality as the final mapping quality
- p Discard paired end reads that are not mapped as correct pairs
- v Output verbose information including base qualities and mapping qualities
- m INT Maximum number of mismatches allowed for a read to be used [7]
- Q INT Maximum allowed number of quality values of mismatches [60]
- q INT Minimum mapping quality allowed for a read to be used [0]
- l FILE File containing the sites at which pileup will be printed out. In this file the first column gives the names of the reference and the second the coordinates. Additional columns will be ignored. [null]
- P also output the base position on the read

Each line consists of chromosome, position, reference base, depth and the bases on reads that cover this position. If -v is added on the command line, base qualities and mapping qualities will be presented in the sixth and seventh columns in order.

The fifth column always starts with '@'. In this column, read bases identical to the reference are showed in comma ',' or dot '.', and read bases different from the reference in letters. A comma or an upper case indicates that the base comes from a read aligned on the forward strand, while a dot or a lower case on the reverse strand.

cns2fq: Extract the consensus sequences in FASTQ format.

```
maq cns2fq [-Q minMapQ] [-n minNeiQ] [-d minDepth] [-D maxDepth] in.cns>out.cns.fastq
```

-Q INT Minimum mapping quality [40]

-d INT Minimum read depth [3]

-n INT Minimum neighbouring quality [20]

-D INT Maximum read dpeth. >=255 for unlimited. [255]

In the sequence lines, bases in lower case are essentially repeats or do not have sufficient coverage; bases in upper case indicate regions where SNPs can be reliably called. In the quality lines, ASCII of a character minus 33 gives the PHRED quality.

cns2snp: Extract SNP sites.

```
maq cns2snp in.cns > out.snp
```

Each line consists of chromosome, position, reference base, consensus base, Phred-like consensus quality, read depth, the average number of hits of reads covering this position, the highest mapping quality of the reads covering the position, the minimum consensus quality in the 3bp flanking regions at each side of the site (6bp in total), the second best call, log likelihood ratio of the second best and the third best call, and the third best call.

The 5th column is the key criterion to judge the reliability of a SNP. However, as this quality is only calculated assuming site independency, the other columns must be considered to get more accurate SNP calls. Script command 'maq.pl SNPfilter' is designed for this.

The 7th column implies whether the site falls in a repetitive region. If no read covering the site can be mapped with high mapping quality, the flanking region is possibly repetitive or in the lack of good reads. An SNP at such site is usually not reliable.

The 8th column roughly gives the copy number of the flanking region in the reference genome. In most cases, this number approaches 1.00, which means the region is about unique. Sometimes you may see non-zero read depth but 0.00 at the 7th column. This indicates that all the reads covering the position have at least two mismatches. Maq only counts the number of 0- and 1-mismatch hits to the reference. This is due to a complex technical issue.

The 9th column gives the neighbouring quality. Filtering on this column is also required to get reliable SNPs. This idea is inspired by NQS, although NQS is initially designed for a single read instead of a consensus.

easyrun: Analyses pipeline for small genomes. Easyrun command will run most of analyses implemented in maq.

```
maq.pl easyrun [-l read1Len] [-d out.dir] [-n nReads] [-A 3adapter] [-e minDep] [-q minCnsQ] [-p] [-2 read2Len] [-a maxIns] [-S] [-N] in.ref.fasta in1.fastq[in2.fastq]
```

OPTIONS

-d DIR	output directory [easyrun]
-n INT	number of reads/pairs in one batch of alignment [2000000]
-S	apply split-read analysis of short indels (maybe very slow)
-N INT	number of haplotypes/strains in the pool (≥ 2) [2]
-A FILE	file for 3'-adapter. The file should contain a single line of sequence [null]
-l INT	length of the first read, 0 for auto [0]
-e INT	minimum read depth required to call a SNP (for SNPfilter) [3]
-q INT	minimum consensus quality for SNPs in cns.final.snp [30]
-p	switch to paired end alignment mode
-2 INT	length of the second read when -p is applied [0]
-a INT	maximum insert size when -p is applied [250]

Several files will be generated in out.dir, among which the following files are the key output:

cns.final.snp	final SNP calls with low quality ones filtered out
cns.fq	consensus sequences and qualities in the FASTQ format

Key commands behind easyrun:

```
maq fasta2bfa ref.fasta ref.bfa;  
maq fastq2bfq part1.fastq part1.bfq;  
maq fastq2bfq part2.fastq part2.bfq;  
maq map part1.map ref.bfa part1.bfq;  
maq map part2.map ref.bfa part2.bfq;  
maq mapmerge aln.map part1.map part2.map;  
maq assemble cns.cns ref.bfa aln.map;
```

3.3.4 Annotation tools

The effect of genetic mutation on phenotype is of significant interest in genetics. Non-synonymous single nucleotide polymorphisms (nsSNP) are genetic mutations that cause a single amino acid substitution (AAS) in a protein sequence. These are capable of affecting the function of the protein, subsequently altering the carrier's phenotype ([Kumar, Henikoff et. al. 2009](#)). Most alterations are deleterious and so are eventually eliminated through purifying selection. However, beneficial mutations can sweep through the population and become fixed, thus contributing to species differentiation. With the massive amounts of genetic variation data being generated from High-throughput sequencing platforms, pinpointing a small subset of functionally important variants is the major challenge. Several annotation pipelines have been developed to identify functionally important variants from the large amount of sequencing data and decipher potential disease causal genes and causal mutations ([Wang et. al., 2010](#)).

3.3.4.1 SIFT: *Sorting Intolerant From Tolerant*

For a given protein sequence, SIFT searches a protein database using PSI-BLAST algorithm to compile a dataset of functionally related protein sequences⁶. It then aligns the homologous sequences with the query sequence. Each position in the alignment is then scanned and probabilities for all possible 20 amino acids are calculated.

These probabilities are normalized by the probability of the most frequent amino acid and are recorded in a scaled probability matrix. Generally, a highly conserved position is intolerant to most substitutions, whereas a poorly conserved position can tolerate most substitutions. If the scaled probability or the SIFT score, lies below a certain threshold value, SIFT predicts a substitution to affect protein function.

SIFT also provides a measure of confidence in the prediction. In case of very little sequence diversity in the set of aligned sequences, may appear as highly conserved which might lead to prediction of neutral substitutions as deleterious. This would thereby result in increase in the false positive error⁷. SIFT calculates a conservation value at each position in the alignment, to assess confidence in the prediction. The conservation value for a position ranges from zero, when all 20 amino acids are observed at that position, to $\log_2 20$ (~4.32), when only one amino acid is observed at that position⁵. To maintain the optimum diversity within the selected sequences, SIFT ensures that the final set of aligned sequences has a median conservation value of ~3.0. If the set of sequences used for prediction are too conserved (median conservation value >3.25), then a low-confidence warning is issued.

SIFT can provide exome-wide analysis of single nucleotide variants and indels.

SIFT_exome_nssnvs.pl (for single nucleotide variants)

SIFT_exome_indels.pl (for indels)

SNPClassifier (for placing variants in genome)

1. SIFT_exome_nssnvs.pl

Input: A list of multiple chromosome coordinates of coding SNVs

Format Description

[chromosome,coordinate,orientation,alleles,user comment(optional)]

Coordinate System:

SIFT accepts both residue-based and a space-based coordinates for single nucleotide variants.

RESIDUE BASED COORDINATE SYSTEM (comma separated)

In a residue based system, each base is assigned a coordinate base on its absolute position, starting from 1.

Format example:

```
2,43881517,1,A/T,#User Comment
2,43857514,1,T/C
6,88375602,1,G/A,#User Comment
22,29307353,-1,T/A
10,115912482,-1,C/T
```

SPACE BASED COORDINATE SYSTEM (comma separated)

The space-based coordinate system counts the spaces before and after bases rather than the bases themselves. Zero always refers to the space before the first base. Space-based coordinates become necessary when describing insertions/deletions and genomic rearrangements.

Format example:

```
2,43881516,43881517,1,A/T,#User Comment
2,43857513,43857514,1,T/C
6,88375601,88375602,1,G/A,#User Comment
22,29307352,29307353,-1,T/A
10,115912481,115912482,-1,C/T
```

Orientation:

It uses 1 for positive strand and -1 for negative strand. If orientation is not known, it makes use of 1 as default.

Alleles:

Use 'base1/base2' where either base1 or base2 may be the reference allele. SIFT will predict for non-reference allele only. For prediction of reference allele, then use base1/base1 where base1 is the reference allele.

Usage:

./SIFT_exome_nssnvs.pl

-i <Query SNP filename with complete path>

-d <Variation db directory path>

-o <Optional: output file with complete path - default=<SIFT_HOME>/tmp>

-m Yes to output multiple transcripts if exists: default No

The following optional parameters can also be entered if the results need to include additional information. They are not included by default

- A 1 to output Ensembl Gene ID
- B 1 to output Gene Name
- C 1 to output Gene Description
- D 1 to output Ensembl Protein Family ID
- E 1 to output Ensembl Protein Family Description
- F 1 to output Ensembl Transcript Status (Known / Novel)
- G 1 to output Protein Family Size
- H 1 to output Ka/Ks (Human-mouse)
- I 1 to output Ka/Ks (Human-macaque)
- J 1 to output OMIM Disease: default
- K 1 to output Allele Frequencies (All Hapmap Populations - weighted average)
- L 1 to output Allele Frequencies (CEU Hapmap population)
- M 1 to output Allele Frequencies (HCB Hapmap population)
- N 1 to output Allele Frequencies (JPT Hapmap population)
- O 1 to output Allele Frequencies (YRI Hapmap population)
- P 1 to output 1000 Genomes Average Allele Frequencies
- Q 1 to output 1000 Genomes European Population Allele Frequencies
- R 1 to output 1000 Genomes East Asian Population Allele Frequencies
- S 1 to output 1000 Genomes West African Population Allele Frequencies
- T 1 to output 1000 Genomes South Asian Population Allele Frequencies
- U 1 to output 1000 Genomes American Population Allele Frequencies

2. SIFT_exome_indels.pl

Input: A list of multiple chromosome coordinates of coding insertion/deletion variants. SIFT scores and predictions are not provided at this stage. It accepts only space-based coordinates for insertion/deletion variants. All values should be in local 0 space based coordinates.

Format Description

[chromosome,coordinate,orientation,alleles,user comment(optional)]

Usage:

./SIFT_exome_indels.pl

- i <Query indels filename with complete path>
- c <coding info directory path>
- d <Variation db directory path>
- o <Optional: output file with complete path - default=<SIFT_HOME>/tmp>

Output:

Amino Acid Position Change: This column contains the change coordinates within the original protein sequence and the modified protein sequence.

Indel location: The percentage indicates the approximate location of the indel in the protein.

Nucleotide change: The input allele (insertion or deletion) and +/- 5 base pairs are shown. For insertions, the inserted bases are displayed in uppercase and the flanking bases are displayed in lowercase. For deletions, the deleted bases are displayed in lowercase whereas the flanking bases are displayed in uppercase.

For example:

Input for insertion variant: "10,102760304,102760304,1,GCGGCT"

Nucleotide change: cggct-GCGGCT-acggc

Input for insertion variant: "12,110521161,110521164,1,/"

Nucleotide change: TGCTG-ctg-TTGCT

Caused Nonsense Mediated Decay: This column indicates whether the input indel is likely to cause NMD.

Repeat detected: This column gets populated if the input insertion/deletion is found to expand or contract a coding repeat region.

3.3.4.2 ANNOVAR

ANNOVAR, functional annotation of genetic variants from high-throughput sequencing data, is an efficient command line Perl program to functionally annotate genetic variants from diverse genomes. ANNOVAR was developed to fill the need and shortlist single nucleotide variants and insertions/deletions, by up-to-date annotation, examining their functional consequence on genes, inferring cytogenetic bands, reporting functional importance scores, finding variants in conserved regions, or identifying variants reported in the 1000 Genomes project and dbSNP.

Given a list of variants with chromosome, start position, end position, reference nucleotide and observed nucleotides, ANNOVAR can perform:

1. **Gene-based annotation:** To identify the affected amino acids and whether SNPs or CNVs cause protein coding changes. Genes from RefSeq, UCSC, ENSEMBL, GENCODE, or many other gene definition systems can be flexibly used.
2. **Region-based annotations:** To identify variants in specific genomic regions, for example, conserved regions among 44 species, predicted transcription factor binding sites, segmental duplication regions, GWAS hits, database of genomic variants, DNase I hypersensitivity sites, ENCODE H3K4Me1/H3K4Me3/H3K27Ac/CTCF sites, CHIP-Seq peaks, RNA-Seq peaks, or many other annotations on genomic intervals.
3. **Filter-based annotation:** To identify variants that are reported in dbSNP, or identify the subset of common SNPs (MAF>1%) in the 1000 Genome Project, or identify subset of non-synonymous SNPs with SIFT score>0.05, or find intergenic variants with GERP++ score<2, or many other annotations on specific mutations.
4. **Other functionalities:** Retrieve the nucleotide sequence in any user-specific genomic positions in batch, identify a candidate gene list for Mendelian diseases from exome data, and other utilities.

Preparation of local annotation databases

ANNOVAR requires "annotation databases" saved in local disk for annotating genetic variants. The `--downdb` argument can be issued to download required annotation database from the UCSC Genome Browser or the ANNOVAR database repository automatically, assuming that the computer is connected to Internet. Several different types of annotation databases can be downloaded, with the command below:

```
annotate_variation.pl -downdb [optional arguments] <table-name> <output-directory-name>
```

1. Download gene annotation databases

Usage:

```
annotate_variation.pl -downdb -buildver hg18 -webfrom annovar refGene humandb
```

The keyword "refGene" tells the program that RefSeq gene-related annotations need to be downloaded.

Besides RefSeq gene, several other gene annotations can be downloaded and used in gene-based annotation, like knownGene gene annotations and ensGene gene annotations.

2. Download region annotation databases from UCSC

Usage:

```
annotate_variation.pl -downdb -buildver hg18 <UCSC Table name> humandb/
```

UCSC Table Name	Explanation
cytoBand	the approximate location of bands seen on Giemsa-stained chromosomes
tfbsConsSites	transcription factor binding sites conserved in the human/mouse/rat alignment, based on transfac Matrix Database (v7.0)
wgRna	snoRNA and miRNA annotations
targetScanS	TargetScan generated miRNA target site predictions
genomicSuperDups	Segmental duplications in genome
phastConsElements*way	Conserved elements produced by the phastCons program based on a whole-genome alignment of vertebrates. Depending on species used, it could be 17way, 28way, 30way, 44way, etc, so users have to specify the *way in the command line argument.
EvoFold	Conserved functional RNA, through RNA secondary structure predictions made with the EvoFold program
Dgv	Database of Genomic Variants, which contains annotations for reported structural variations
omimGene	Canonical UCSC genes that have been associated with identifiers in the Online Mendelian Inheritance in Man (OMIM) database. As advised by UCSC, the results "should be treated with skepticism and any conclusions based on them should be carefully scrutinized using independent resources", including manual inspection of primary literature.
gwasCatalog	Published GWAS results on diverse human diseases.

(other)

All other databases, using the URL `ftp://hgdownload.cse.ucsc.edu/goldenPath/<build-version>/database/<table-name>.txt.gz`, where `<build-version>` and `<table-name>` is specified by the user. If the Table does not exist in UCSC databases, an error will be thrown by the program.

3. Users can supply additional filter-based annotation databases

In addition to downloading annotation databases from Internet, several types of self-annotated databases can be supplied. The "generic" format can be used for filter-based annotations, while the "gff3" format can be used for region-based annotations or gene-based annotations.

Table	Dataset	Explanation
generic	any filter-based data set conforming to generic format (for use with --filter operation)	Users can generate their own variants databases with the simple format (chr, start, end, reference allele, observed allele, and any other columns), and ANNOVAR can process this database using -dbtype generic argument. For example, some users may want to compute whole-exome PolyPhen scores and use ANNOVAR to annotate variants using these scores.
gff3	any annotation data set conforming to Generic Feature Format 3 (GFF3), a current golden standard for model-organism sequence feature annotations (for use with -regionanno operation)	Users can supply a GFF3 formatted database file, and annovar will perform region-based annotations on query against this file. A detailed description on GFF3 format can be found at sequence ontology website: http://www.sequenceontology.org/gff3.shtml . It has become the standard for many model organism databases for sequence feature exchange, so essentially users have unlimited ability to annotate their variants, as long as a particular annotation database exist in GFF3 format.
vcf	any <i>custom</i> VCF file with population frequency data on alleles	VCF format is adopted by the 1000 Genomes Project to present variation data. The file may contain called alleles and their frequencies in a population, but may also contain individual genotypes for each subject in a population. ANNOVAR will examine the annotated mutations in a population.
bed	a BED file with chr, start and end position	Users can supply a custom BED file for region-based annotation. For example, after an exome sequencing experiments you generated variant calls, but are only interested in the calls located in the "target region" of the exome enrichment array; in this case, you can use the BED file provided by array manufacturer to filter the subset of variants located within target regions.

Standard format of ANNOVAR input file

ANNOVAR takes text-based input files, where each line corresponds to one variant. On each line, the first five space- or tab- delimited columns represent chromosome, start position, end position, the reference nucleotides and the observed nucleotides. Additional columns can be supplied and will be printed out in identical form. For convenience, users can use:

“0” To fill in the reference nucleotides, if this information is not readily available

“_” To represent a null nucleotide, insertions, deletions or block substitutions can be readily represented by this simple file format

By default, 1-based coordinate system will be assumed; if --zerostart argument is issued, a half-open zero-based coordinate system will be used in ANNOVAR instead.

Example input file:

```
1 161003087 161003087 C T comments: rs1000050, a SNP in Illumina SNP arrays
1 84647761 84647761 C T comments: rs6576700 or SNP_A-1780419, a SNP in Affymetrix SNP arrays
1 13133880 13133881 TC - comments: rs59770105, a 2-bp deletion
1 11326183 11326183 - AT comments: rs35561142, a 2-bp insertion
1 105293754 105293754 A ATAAA comments: rs10552169, a block substitution
1 67478546 67478546 G A comments: rs11209026 (R381Q), a SNP in IL23R associated with Crohn's disease
```

Format conversion script: To generate ANNOVAR input files

The *convert2annovar.pl* script provides some very rudimentary utility to convert other "genotype calling" format into ANNOVAR format. Currently, the program can handle Samtools genotype-calling pileup format, Illumina export format from GenomeStudio, SOLiD GFF genotype-calling format, Complete Genomics variant format, and VCF format.

Usage:

```
convert2annovar.pl -format <format> -out <output filename> <input filename>
```

Options:

- format cg: To convert Complete Genomics genotyping calling format to ANNOVAR format
- format gff3-solid: To convert GFF3-SOLiD format to ANNOVAR format
- format soapsnp: To convert SOAPSnp format to ANNOVAR format
- format maq: To convert MAQ genotype calling format to ANNOVAR format
- format cassava: To convert CASAVA genotype calling format to ANNOVAR format

OUTPUT

Two output files are generated: *.variant_function and *.exonic_variant_function

Output file 1 (refSeq gene annotation)

The first file contains annotation for all variants, by adding two columns to the beginning of each input line.

The first column tells whether the variant hit exons or intergenic regions, introns or non-coding RNA genes.

If the variant is exonic/intronic/ncRNA, the second column gives the gene name (if multiple genes are hit, comma will be added between gene names); if not, the second column will give the two neighboring genes and the distance to these neighboring genes.

The possible values of the first column are summarized below:

Value	Default precedence	Explanation
exonic	1	variant overlaps a coding exon
splicing	1	variant is within 2-bp of a splicing junction (use -splicing_threshold to change this)
ncRNA	2	variant overlaps a transcript without coding annotation in the gene definition (see Notes below for more explanation)
UTR5	3	variant overlaps a 5' untranslated region
UTR3	3	variant overlaps a 3' untranslated region

intronic	4	variant overlaps an intron
upstream	5	variant overlaps 1-kb region upstream of transcription start site
downstream	5	variant overlaps 1-kb region downstream of transcription end site (use -neargene to change this)
intergenic	6	variant is in intergenic region

To know all the functional consequences, --separate argument should be used. In this case, several output lines may be present for each variant, representing several possible functional consequences.

Output file 2 (refSeq gene annotation)

The second output file, ex1.human.exonic_variant_function, contains the amino acid changes as a result of the exonic variant. The exact format of the output below may change slightly between different versions of ANNOVAR. Only exonic variants are annotated in this file.

The first column gives the line # in the original input file.

The second field tells the functional consequences of the variant (possible values in this fields include: nonsynonymous SNV, synonymous SNV, frameshift insertion, frameshift deletion, nonframeshift insertion, nonframeshift deletion, frameshift block substitution, nonframeshift block substitution).

The third column contains the gene name, the transcript identifier and the sequence change in the corresponding transcript. A [standard nomenclature](#) is used in specifying the sequence changes (you may want to add -hgvs argument so that the cDNA level annotation is compatible with HGVS nomenclature).

More detailed explanation of these exonic_variant_function annotations are given below.

Annotation	Precedence	Explanation
frameshift insertion	1	an insertion of one or more nucleotides that cause frameshift changes in protein coding sequence
frameshift deletion	2	a deletion of one or more nucleotides that cause frameshift changes in protein coding sequence
frameshift block substitution	3	a block substitution of one or more nucleotides that cause frameshift changes in protein coding sequence
stopgain	4	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block substitution that lead to the immediate creation of stop codon at the variant site. For frameshift mutations, the creation of stop codon downstream of the variant will not be counted as "stopgain"!
stoploss	5	a nonsynonymous SNV, frameshift insertion/deletion, nonframeshift insertion/deletion or block substitution that lead to the immediate elimination of stop codon at the variant site
nonframeshift insertion	6	an insertion of 3 or multiples of 3 nucleotides that do not cause frameshift changes in protein coding sequence
nonframeshift deletion	7	a deletion of 3 or multiples of 3 nucleotides that do not cause frameshift changes in protein coding sequence

nonframeshift block substitution	8	a block substitution of one or more nucleotides that do not cause frameshift changes in protein coding sequence
nonsynonymous SNV	9	a single nucleotide change that cause an amino acid change
synonymous SNV	10	a single nucleotide change that cause an amino acid change

3.3.5 Existing Databases

3.3.5.1 *dbSNP*

The NCBI Short Genetic Variations (SNV) database, also known as dbSNP, catalogs short variations in nucleotide sequences from a wide range of organisms. These variations include single nucleotide variations, short nucleotide insertions and deletions, short tandem repeats and microsatellites. SNVs may be common, thus representing true polymorphisms or they may be rare. Each dbSNP entry includes the sequence context of the polymorphism (i.e., the surrounding sequence), the occurrence frequency of the polymorphism (by population or individual), and the experimental method(s), protocols, and conditions used to assay the variation (Kitts A. and Sherry S., 2011). It hosts 40,564 entries for *Mycobacterium tuberculosis*, which includes SNPs as well as INDELS.

NCBI dbSNP Short Genetic Variations

PubMed Nucleotide Protein Genome Structure PopSet Taxonomy OMIM Books SNP

Search for SNP on NCBI Reference Assembly

Search Entrez SNP for Go

Have a question about dbSNP? Try searching the SNP FAQ Archive! Go

ANNOUNCEMENT

04/25/2012: RELEASE: NCBI dbSNP Build 138 Phase I

Component Availability Dates:

Component	Date Available
dbSNP Web Query	April 25, 2013

GENERAL

HUMAN VARIATION

Search, Annotate, Submit

Annotate and Submit Batch Data with Clinical Impact

Attributes for Filtering Variation

SNP SUBMISSION

DOCUMENTATION

SEARCH

RELATED SITES

Search by IDs on All Assemblies

Note: *rs#* and *ss#* must be prefixed with "rs" or "ss", respectively (i.e. rs25, ss25)

ID: Reference cluster ID(rs#)

Search Reset

Submission Information

- [By Submitter](#)
- [New Submitted Batches](#)
- [Method](#)
- [Population](#)
- [Publication](#)

Batch

- Enter List
 - [NCBI Assay ID\(ss\)](#)
 - [Reference SNP ID\(rs\)](#)
 - [Local SNP ID](#)
- Upload List
 - [NCBI Assay ID\(ss\)](#)
 - [Reference SNP ID\(rs\)](#)
 - [Local SNP ID](#)

[Batch Query Help](#)

Fig. 10: Home page of dbSNP

3.3.5.2 TBDB

The Tuberculosis Database (TBDB) is an integrated database providing access to TB genomic data and resources, relevant to the discovery and development of TB drugs, vaccines and biomarkers. The current release of TBDB houses genome sequence data and annotations for 28 different *Mycobacterium tuberculosis* strains and related bacteria. TBDB stores pre- and post-publication gene-expression data from *M. tuberculosis* and its close relatives. TBDB currently hosts data for nearly 1500 public tuberculosis microarrays and 260 arrays for *Streptomyces*. In addition, TBDB provides access to a suite of comparative genomics and microarray analysis software. By bringing together *M. tuberculosis* genome annotation and gene-expression data with a suite of analysis tools, TBDB (<http://www.tbdb.org/>) provides a unique discovery platform for TB research (Reddy *et. al.*, 2009).

TB Database

AN INTEGRATED PLATFORM FOR TUBERCULOSIS RESEARCH

QUICK SEARCH

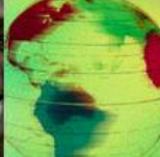
Sign In

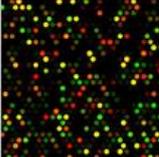
Feedback

Home
Publications
Expression Data
Genomic Data
Systems Biology
Proteins
Advanced Search
Help

About
Partners
TBDB People
Announcements Archive
News and Updates Archive
TB Community
Site Map






Announcements

January 2013

Workshop Announcement: Systems Biology of TB' workshop with your NIH consortium at the Keystone Meeting, on 17 March 2013.

August 2012

We have moved half of TBDB to the Amazon Cloud. There are a few known problems we are working on:

- GenePattern has occasional anomalous behavior.
- A facility for users to upload their data has not been implemented yet.

Please report any problems or anomalies you encounter - missing files, broken links, etc. Thank you for your continued support.

[Announcements Archive...](#)

Quick Search

Enter gene name, gene sequence name, author name, title, or keyword.

Examples: RV2429, cobS, Schoolnik, log phase. The search term is case insensitive.

Tutorials

For help, please select one of our tutorials below or look at our FAQ page.

All Tutorials

Publications

Expression

Genomic Data

How To...

Search

Accessing Data

Gene Profiles

Samples & Conditions

Finding Genes

Gene Details

Finding Orthologs

Find the DosR Regulon

Find the RD1 Region

These are a small sample of the tutorials currently available. Push the 'All Tutorials' button to view all of the tutorials.

TBDatabase (TBDB) makes available the tools and resources available at the Stanford Microarray Database and the Broad Institute.

Fig. 11: Home Page of TBDB

3.3.5.3 MTCID

MTCID (*M. tuberculosis* clinical isolate genetic polymorphism database) is an attempt to provide a comprehensive repository to store, access and disseminate single nucleotide polymorphism (SNPs) and spoligotyping profiles of *M. tuberculosis*. It can be used to automatically upload the information available with a user that adds to the existing database at the backend. Besides it may also aid in maintaining clinical profiles of TB and treatment of patients. The database has 'search' features and is available at <http://ccbb.jnu.ac.in/Tb> (Bharti *et. al.*, 2012).

HOME>>
SUBMIT
SEARCH
LINKS>>
REFERENCES
CONTACT US

Query

Subject

Output SNP

To submit the mutation in your query sequence, [click here](#).

© School of Computational and Integrative Sciences, JNU, New Delhi

Fig. 12: Search Page of MTCID

3.3.5.3 TbDReaMDB

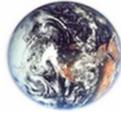
TbDReaMDB (<http://www.tbdreamdb.com/index.html>) is a comprehensive resource on drug resistance mutations in *M. tuberculosis*. The resource has been compiled by conducting a systematic review to identify drug resistance mutations from the existing literature to include in the database. For each mutation, the database provides complete codon changes for each mutation at both the nucleotide and amino acid level. The database is divided into two parts. The first part lists all the unique mutations reported in drug-resistant TB isolates, as well as information on the time period of isolate collection, country of origin, molecular detection method, resistance pattern, and susceptibility testing method. As of September 1, 2008, TbDReaMDB contains 946 unique mutations associated with seven different drug classes and spread over 36 genes, two intergenic/promoter regions, and one ribosomal RNA coding region. The second part of the database provides data on the relative frequency of the most common mutations associated with resistance to specific drugs, as reported in surveys from diverse geographical sites (Sandgren *et. al.*, 2009).



TB Drug Resistance Mutation Database

AMI | EMB | ETH | FLQ | INH | PAS | PZA | RIF | SM | **Information**

Select drug class by clicking on the tabs above



Tuberculosis is a global health threat

Tuberculosis (TB) remains the leading cause of death in the world from a preventable and curable infectious disease. The emergence and spread of anti-TB drug resistance has developed into a serious threat to the global TB control plans. Standard methods to diagnose drug resistant TB rely on culture and phenotypic drug susceptibility testing (DST). Since *M. tuberculosis* grows slowly, routine DST can take weeks to months, delays during which patients receive suboptimal therapy that may lead to development of additional resistance and further spread of drug resistant TB.



Novel diagnostic tools are needed

New tools for rapid diagnosis of drug resistant TB are urgently needed. Sequence based diagnostic methods are being developed that detect specific mutations associated with drug resistance; these tools have the advantage of being rapid, high-throughput, and easily compared between laboratories. The development of such diagnostic tools relies on information about mutations that lead to drug resistance and evaluations of the relative frequency of these specific mutations.



Mutations confer drug resistance

In response to this urgent need to improve diagnostics of resistant TB, we have compiled a comprehensive list of the genetic polymorphisms associated with first and second line drug resistance in clinical *M. tuberculosis* isolates throughout the world. We have reviewed the most common mutations found for the major groups of anti-tuberculosis drugs, establishing a database that we hope will enable the development of sequence based tools for diagnosis and surveillance of drug resistance in tuberculosis.

THE DATABASE WAS LAST UPDATED IN APRIL 2010

Fig. 13: Home Page of TBDReaMDB

4. METHODOLOGY

4.1 Datasets and Methods

Genome Datasets: The availability of re-sequencing data-sets in public domain was extensively used to compile the *M. tuberculosis* variome. We retrieved a total of 37 datasets from re-sequencing projects of *M. tuberculosis* from the NCBI Sequence Read Archive (SRA). These data sets correspond to a total of 469 strain samples. All datasets were part of re-sequencing projects using next-generation sequencing. Only datasets in public domain and not in embargo were considered for our analysis.

The Sequence Read Archive (SRA) stores raw sequencing data from next-generation sequencing platforms including Applied Biosystems SOLiD® System, Complete Genomics®, Helicos Heliscope®, Illumina Genome Analyzer®, Pacific Biosciences SMRT®, and Roche 454 GS System®. The SRA is the single best resource for useful data from initiatives such as the 1,000 Genomes Project and institutions like the Broad Institute, Washington University, and the Wellcome Trust Sanger Institute. Sequencing reads from 469 strain samples were downloaded from SRA in .sra format and saved separately in different folders with the corresponding sample names. This was done so that multiple reads from the same sample can be easily aligned and used to derive a consensus base-call at each position.

The retrieval and segregation of such a large dataset was done computationally using a perl script. This was then followed by conversion of .sra format to the maq acceptable .fastq format using sratoolkit.

Usage:

```
.fastq-dump -split-files -O <output directory> <path>
```

OPTIONS

```
--split-files    Dump each read into a separate file.  
                  Files will receive suffix corresponding to read number.  
-O              Output directory, default is '.'
```

The H37Rv reference genome (NC_000962.2) was used as the reference for mapping the reads, since it is considered to be best characterized strain of *Mycobacterium tuberculosis*. The genome comprises 4,411,529 base pairs, contains around 4,000 genes, and has a very high guanine + cytosine content that is reflected in the biased amino-acid content of the proteins (Cole *et. al.*, 1998).

4.2 Read Mapping and variant calling

We used a popular and extensively used quality aware reference mapping toolkit Mapping and Assembly with Quality (MAQ)([Li et. al., 2008](#)). To assure good quality of mapping base-wise mean quality of reads was first deduced using FastQC tool (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>). The reads were then filtered based on the quality score. For each run, read length greater than 15 with mean Phred quality more than 20 were selected for reference alignment while rest of the datasets were discarded. The consensus mapping quality was also used to weed out low quality assemblies. Further all data sets which did not match quality criteria of mean Phred score cutoff of 20 across the genome were dropped from further analysis. The resulting dataset comprised of a total of 469 unique samples which were then mapped to the reference genome. The variants were called using the MAQ variant caller using parameters described as follows:

Parameter for single end runs

```
$ maq.pl easyrun -d <output folder> -l <length of read> -e 10 -q 60 <reference> <read>
```

Parameter for paired end runs

```
$ maq.pl easyrun -d <output folder> -l <length of forward read> -2 <length of reverse read> -e 10 -q 60 -p <ref> <read1> <read2>
```

Easyrun command of MAQ has been used with options `-d`, `-l`, `-e` and `-q` to mention the output directory, read length, minimum read depth to call an SNP and quality threshold for the final SNP calls respectively. Reads from all the 469 samples were mapped on to the H37Rv Reference genome sequentially using a perl script.

4.3 Variant comparison

We retrieved variants from dbSNP and MTCID for comparisons. The databases briefly had 3,885 & 263 variants annotated respectively on the *M. tuberculosis* H37Rv genome. MTCID is a repository providing access to the genetic polymorphisms from clinical isolates and also provides information on their strains and associated spoligotypes ([Bharti et. al., 2012](#)), while dbSNP archives genomic variants of organisms. Additionally, the datasets corresponding to drug-resistance traits were retrieved from TB Drug resistance mutation database (TBDreamDB) ([Sandgren et. al., 2009](#)). This dataset comprised of over 1,100 variants corresponding to about 40 genes for 9 antibiotics. The comparison of the variants generated from re-sequencing datasets to the known variants reported was done using *custom* scripts.

4.4 Variant annotation

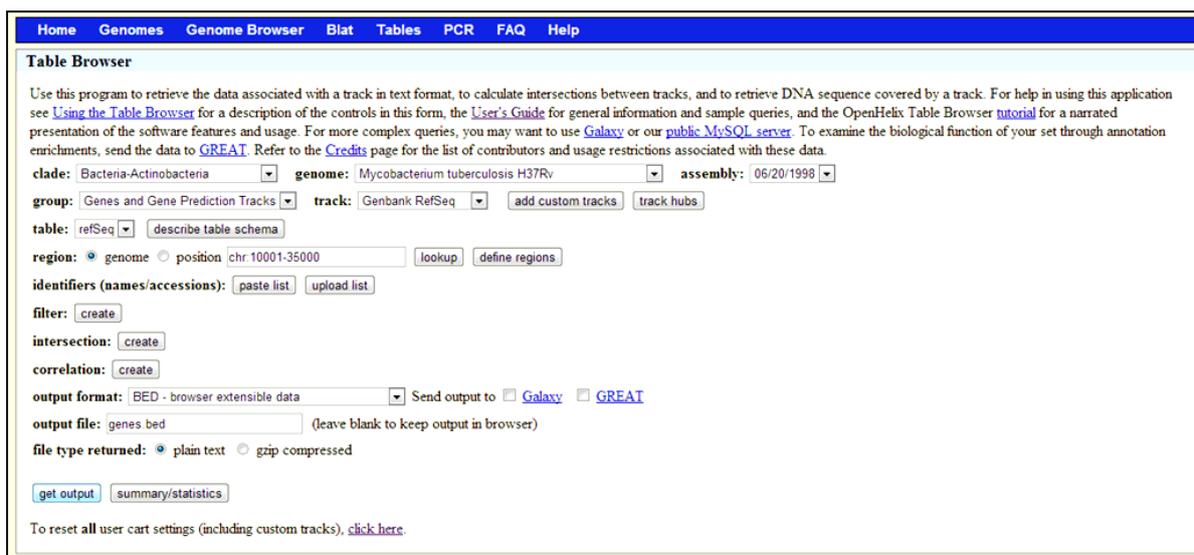
The variants were annotated using the popular variant annotation toolkit ANNOVAR ([Wang et. al., 2010](#)). Briefly the gene coordinates were formatted and fed to ANNOVAR using custom scripts and the variants were annotated in a number of parameters like gene loci (genic, intergenic), effect (synonymous, non-synonymous, stop gain/loss) etc. Gene based annotation was done using the input file that was prepared using a custom script.

```
annotate_variation.pl -geneanno -dbtype knowngene mtb.txt mtbdb/ -out annovar_out
```

where mtb.txt is the maq output variant file in annovar input format, mtbdb is the manually prepared database and annovar_out is the output directory wherein the variant_function and exonic_variant_function files created by ANNOVAR are saved. The annotation in these files was then mapped onto the variant list deciphered by MAQ with respect to the variant position.

4.5 Mapping Genes onto the Variants

To download genic information, for the H37Rv reference genome, UCSC genome browser was used. In the table browser Bacteria-Actinobacteria was chosen as clade, Mycobacterium tuberculosis H37Rv as genome, group was set as Genes and Gene prediction Tracks and Track was set to Genbank RefSeq. With these settings a file named genes was downloaded in BED format. Genes.bed file was then used to map the variants with the gene related information if the variant positions fall within the gene loci using a perl script (map_gene.pl-APPENDIX-I). The output was saved in a file with the name of map_exonic_gene.txt. Once the genes were mapped onto the exonic variants, a file was compiled with both the exonic as well as non-exonic variants named map_all_var_gene.txt



The screenshot shows the UCSC Table Browser interface. The navigation bar includes links for Home, Genomes, Genome Browser, Blat, Tables, PCR, FAQ, and Help. The main content area is titled "Table Browser" and contains the following configuration options:

- clade:** Bacteria-Actinobacteria
- genome:** Mycobacterium tuberculosis H37Rv
- assembly:** 06/20/1998
- group:** Genes and Gene Prediction Tracks
- track:** Genbank RefSeq
- table:** refSeq
- region:** genome (selected), position: chr.10001-35000
- identifiers (names/accessions):** paste list, upload list
- filter:** create
- intersection:** create
- correlation:** create
- output format:** BED - browser extensible data
- Send output to:** Galaxy (checked), GREAT (unchecked)
- output file:** genes.bed
- file type returned:** plain text (selected), gzip compressed (unchecked)

Buttons for "get output" and "summary/statistics" are visible at the bottom. A note at the bottom states: "To reset all user cart settings (including custom tracks), [click here](#)."

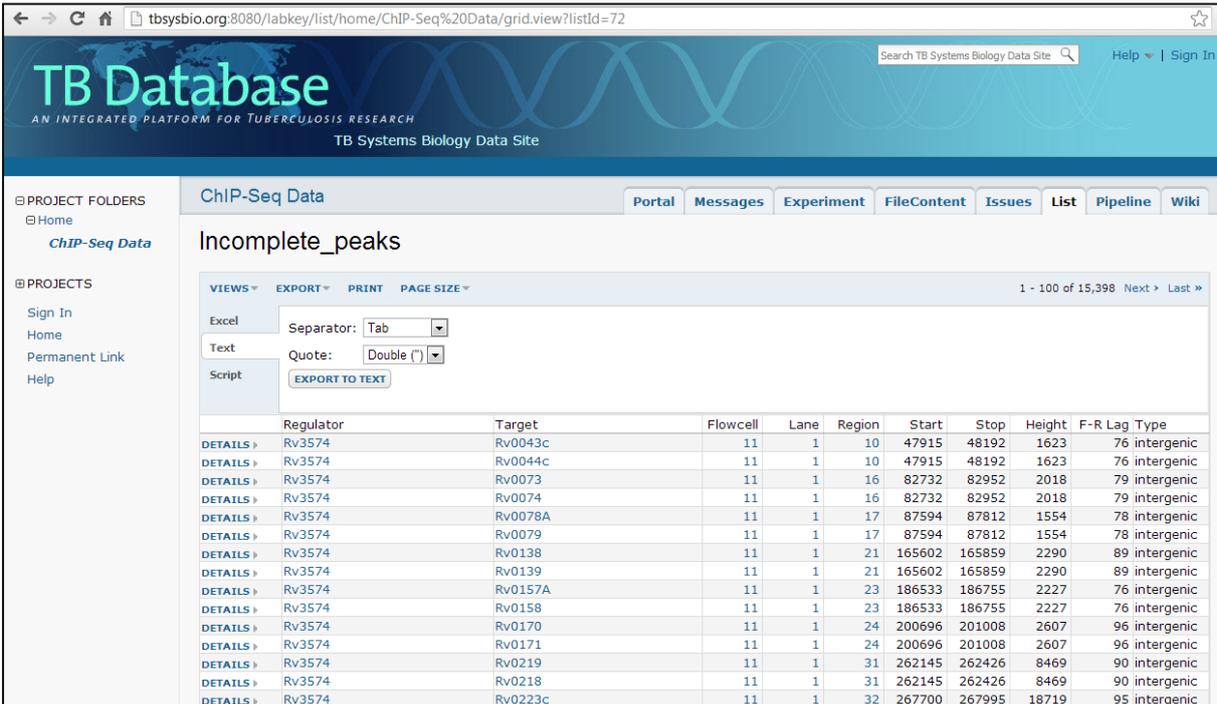
Fig. 14: Downloading MTB gene table from UCSC Table Browser

4.6 Functional analyses of variations

The genic non-synonymous variants were analyzed further using Sorting Intolerant from tolerant (SIFT) (Ng and Henikoff, 2001) for potential functional consequences. Briefly the annotations from ANNOVAR were reformatted using *custom* scripts and used as template for analysis using SIFT. For this input file was first prepared in the Residue based format and SIFT_exome_nssnvs.pl command was used to notate the variants with functional consequences. The output file of this tool was again fed to a perl script to map functional consequence onto the exonic as well as non-exonic variants.

4.7 Mapping of variants to regulatory regions

The recent availability of ChIP-seq datasets for *M. tuberculosis* was extensively used for the mapping of variants. These data sets were retrieved from the Broad Institute repository and correspond to ChIP-seq peaks corresponding to 50 transcription factors. The respective positions of the peaks were downloaded and variants were matched to the respective loci using *custom* scripts (map_peaks.pl – APPENDIX-II) and saved as map_var_gene_peaks.txt



The screenshot shows the TB Database ChIP-Seq Data export interface. The page title is "Incomplete_peaks". The interface includes a navigation menu on the left with "PROJECT FOLDERS" (Home, ChIP-Seq Data) and "PROJECTS" (Sign In, Home, Permanent Link, Help). The main content area has tabs for "Portal", "Messages", "Experiment", "FileContent", "Issues", "List", "Pipeline", and "Wiki". Below the tabs, there are options for "VIEWS", "EXPORT", "PRINT", and "PAGE SIZE". The "EXPORT" section has a "Separator" dropdown set to "Tab" and a "Quote" dropdown set to "Double (")". An "EXPORT TO TEXT" button is visible. Below the export options is a table with the following columns: Regulator, Target, Flowcell, Lane, Region, Start, Stop, Height, F-R Lag, and Type. The table contains 15 rows of data, each with a "DETAILS" link to the left of the "Regulator" column.

	Regulator	Target	Flowcell	Lane	Region	Start	Stop	Height	F-R Lag	Type
DETAILS	Rv3574	Rv0043c	11	1	10	47915	48192	1623	76	intergenic
DETAILS	Rv3574	Rv0044c	11	1	10	47915	48192	1623	76	intergenic
DETAILS	Rv3574	Rv0073	11	1	16	82732	82952	2018	79	intergenic
DETAILS	Rv3574	Rv0074	11	1	16	82732	82952	2018	79	intergenic
DETAILS	Rv3574	Rv0078A	11	1	17	87594	87812	1554	78	intergenic
DETAILS	Rv3574	Rv0079	11	1	17	87594	87812	1554	78	intergenic
DETAILS	Rv3574	Rv0138	11	1	21	165602	165859	2290	89	intergenic
DETAILS	Rv3574	Rv0139	11	1	21	165602	165859	2290	89	intergenic
DETAILS	Rv3574	Rv0157A	11	1	23	186533	186755	2227	76	intergenic
DETAILS	Rv3574	Rv0158	11	1	23	186533	186755	2227	76	intergenic
DETAILS	Rv3574	Rv0170	11	1	24	200696	201008	2607	96	intergenic
DETAILS	Rv3574	Rv0171	11	1	24	200696	201008	2607	96	intergenic
DETAILS	Rv3574	Rv0219	11	1	31	262145	262426	8469	90	intergenic
DETAILS	Rv3574	Rv0218	11	1	31	262145	262426	8469	90	intergenic
DETAILS	Rv3574	Rv0223c	11	1	32	267700	267995	18719	95	intergenic

Fig. 15: Exporting of ChIP-Seq peaks data from TBDB

4.8 Mapping of variants to ncRNA

Similar to retrieving genic information from UCSC genome browser, information for ncRNA was also retrieved to identify the variants that fall within ncRNAs. In the table browser Bacteria-Actinobacteria was chosen as clade, Mycobacterium tuberculosis H37Rv as genome, group was set as Genes and Gene prediction Tracks, Track was set to GenBank ncRNAs and Table as gbRNAs. With these settings a file named nc_rna was downloaded and

was then used to map the variants with the ncRNA related information if the variant position falls within the loci using a perl script (map_ncrna.pl-APPENDIX-III).



Fig. 16: Downloading ncRNA data from UCSC Table Browser

4.9 Mapping of variants to Sample information

Sample information was extracted from SRA, with respect to the Sample Accession number, Experiment Accession Numbers corresponding to those samples and Accession number for the corresponding Studies, for the identified variants. Another perl script as then designed to add Sample, Experiment and Study information to the map_var_gene_peaks.txt file.

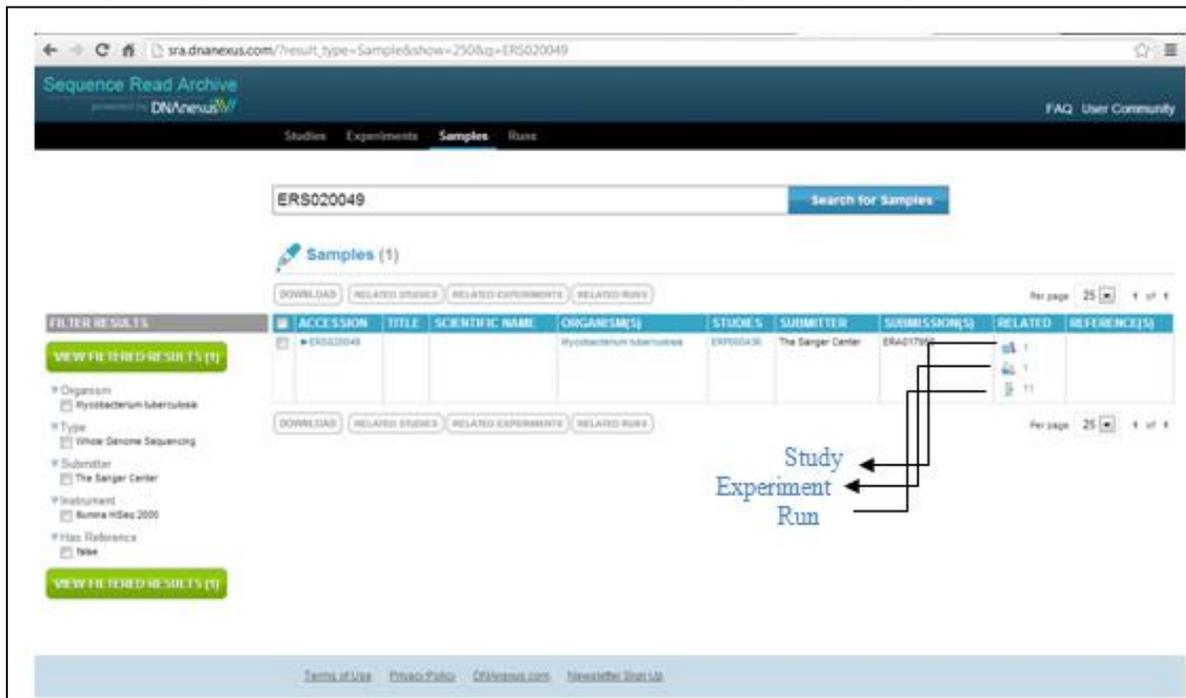


Fig. 17: Extracting strain information from Sequence Read Archive

4.10 Mapping of variants to information related to Drug Resistance

The datasets corresponding to drug-resistance traits were retrieved from TB Drug resistance mutation database (TBDreamDB - http://www.tbdreamdb.com/TBDRemMDB_HighConfidenceMutations201004.txt) which comprises over 1,100 variants corresponding to about 40 genes for 9 antibiotics. Variant positions in the file composed until now were cross-checked for drug resistance from the data retrieved from TBDreamDB. In case of any match found, the file was updated with the corresponding information.

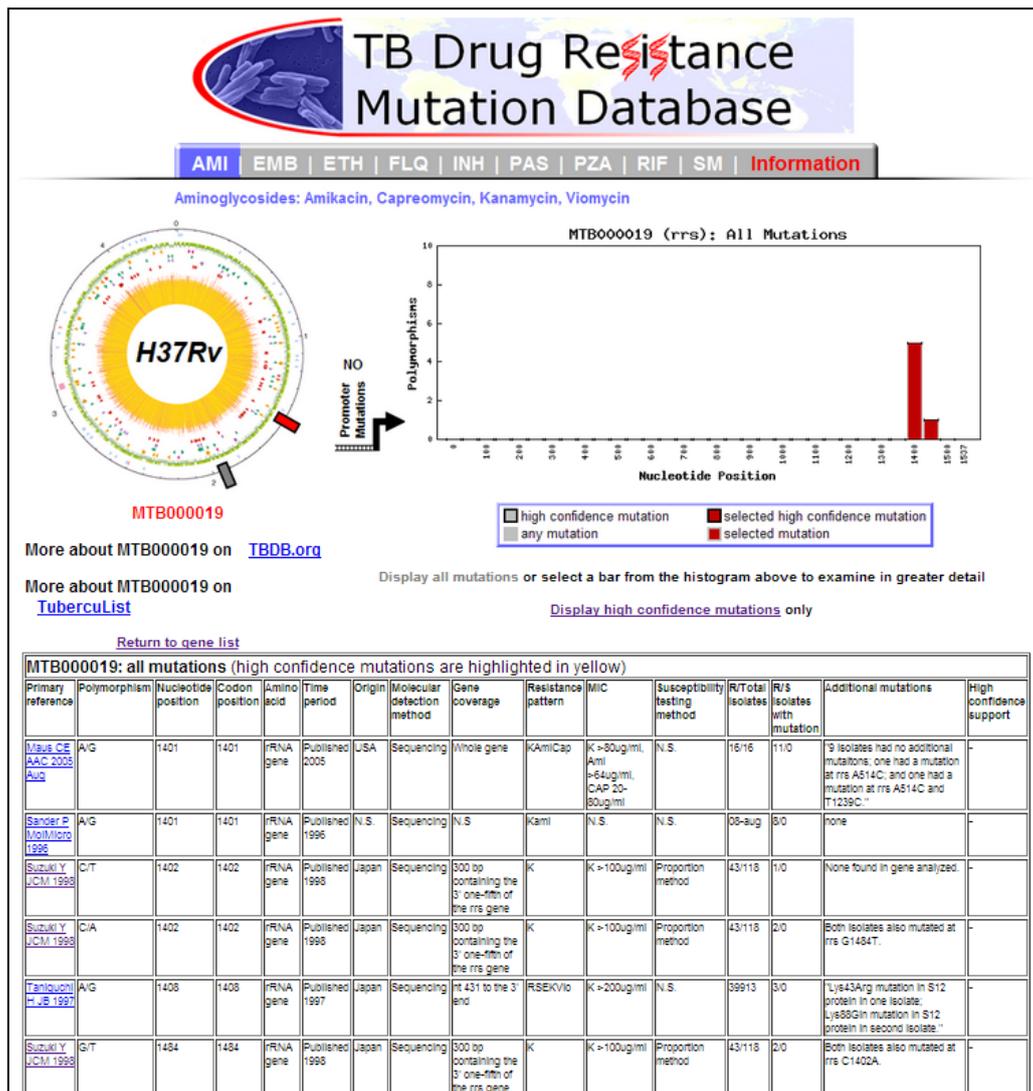


Fig. 18: Extracting the drug resistance data from TBDreamDB

4.11 Mapping of variants to those available in existing databases

Entire variant dataset available in existing databases like dbSNP, MTCID and TBDB were retrieved. These were then cross-checked with our own variant dataset. In case of a match found the variant information was appended with the database name. This was done so as to inform whether a variation has been already reported or not and if so then in which database.

4.12 Database construction

With the variant information compiled as mentioned above, a database was constructed using MySQL, an Open source relational database system. The data model for the same was created using MySQL Workbench, which is a visual database design application that can be used to efficiently design, manage and document database schemata. To create a new data model, the following steps were performed:

1. Under the heading “Data modeling”, “Create new data model” was selected.
2. Double-clicked on the “mydb” tab to enter the name of the database.
3. Double-clicked on “Add table” to add a new table to the database. The table name, column names and types as well as declaration of primary key, foreign keys and other constraints and triggers, was done here.

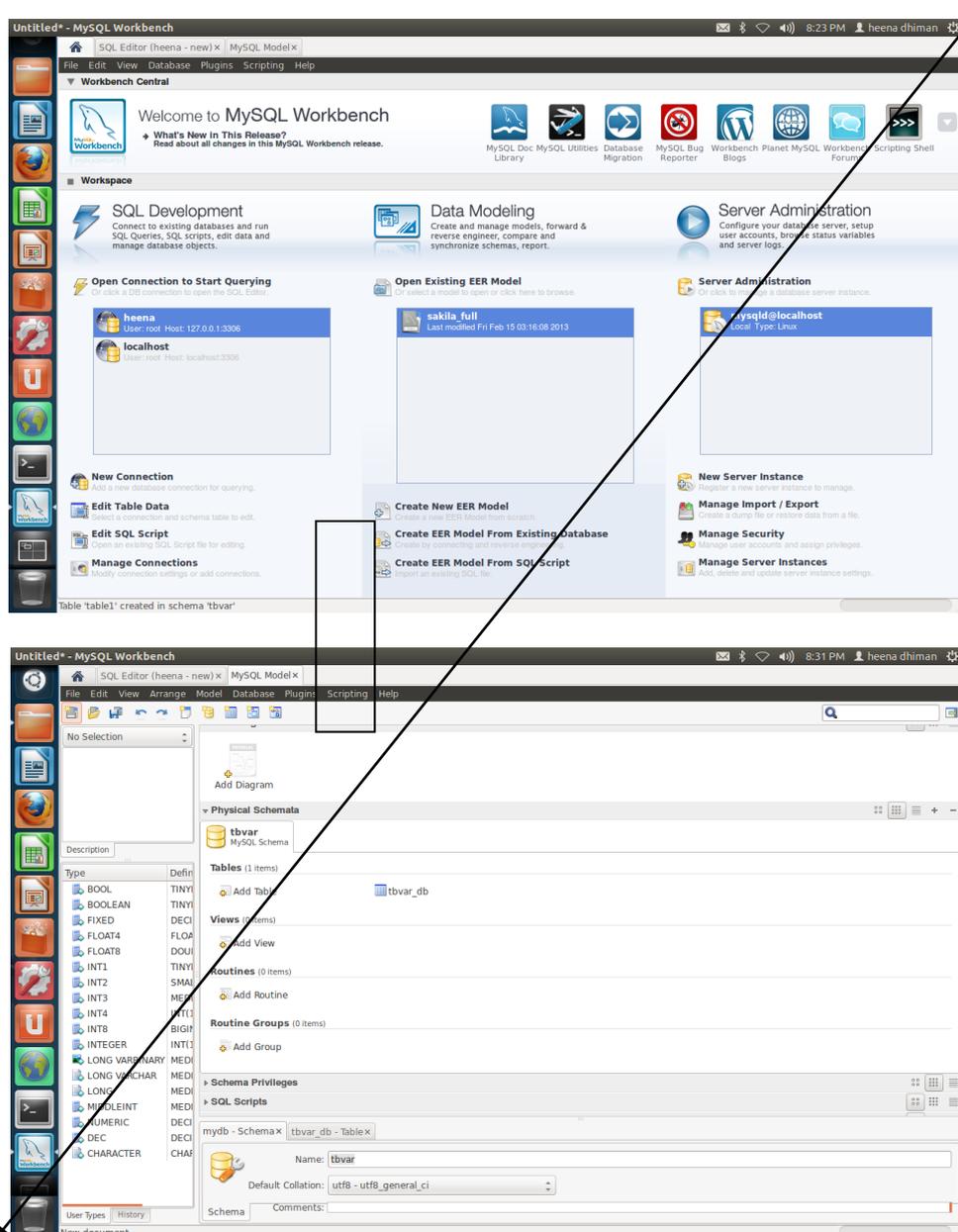


Fig. 19: Constructing a database on MySQL Workbench

To create a new table:

1. Double-clicked on “Add table” under the heading Tables.
2. Typed the table name in the text box.
3. Double-clicked under the heading “Column Name” and typed the name of the column that is the primary key. It was ensured that the PK (Primary Key) and NN (Not Null) checkboxes were selected.
4. Selected the data-type of the column.
5. Created the other fields in the same way.

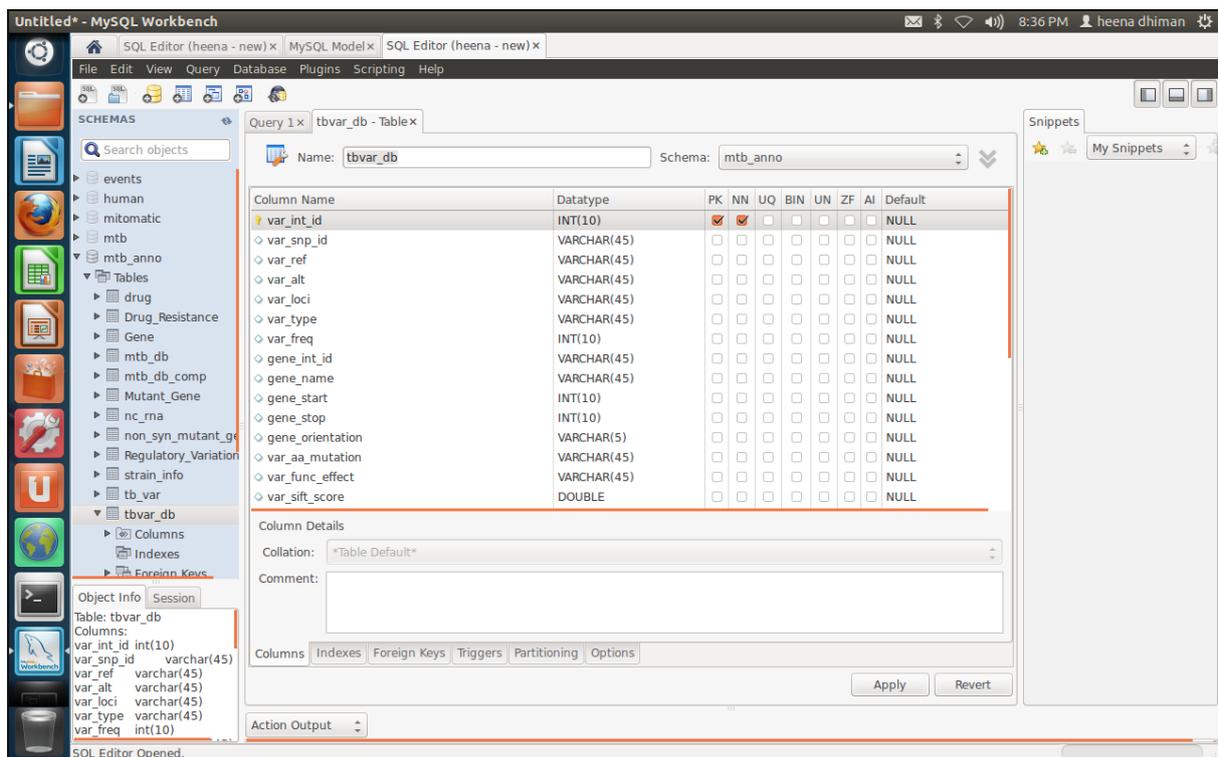


Fig. 20: Developing a table with required columns on MySQL Workbench

The data file was uploaded into MySQL command-line using the following command on the terminal:

```
mysql -h localhost -u root -p matb_anno -local -infile  
mysql> Load data local -infile tbvar.txt into table tbvar_db
```

Interfaces were coded in Perl-CGI and HTML/CSS and the server was hosted on Apache HTTP server. The Common Gateway Interface, or CGI, is a set of standards that define how information is exchanged between the web server and a *custom* script.

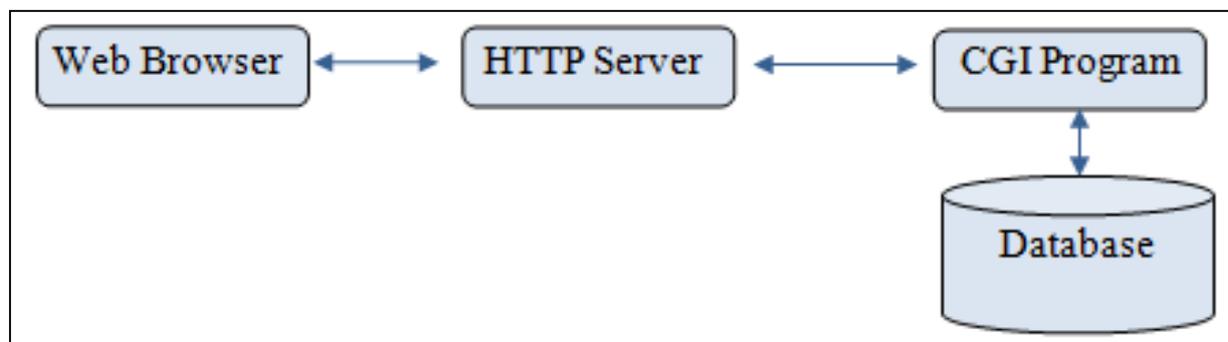


Fig. 21: Information retrieval from database by web browser using CGI

The process of retrieving data from an online database through a web browser is discussed as follows:

1. The browser sends the query to the server using the GET or POST methods.
2. The server executes the CGI program which connects to the database and fetches the required information.
3. This retrieved information is passed to the server.
4. The server then sends the information back to the browser where it is displayed to the user.

Table III: Description of HTML scripts used to develop the web interface of tbvar

S. No.	Script Name	Description
1.	index.html	The home page wherein user can input query in form of variant location, range, gene-name or rVID
2.	annoTB.html	To annotate list of variations together
3.	contact.html	Feedback form
4.	browser.html	An access to the <i>Mycobacterium tuberculosis</i> genome browser
5.	help.html	An access to the user manual

Table IV: Description of Perl-CGI scripts used to develop the web interface of tbvar

S. No.	Script Name	Description
1.	tbvar.cgi	To extract the input query by the user, retrieve the corresponding information from the compiled datasheet and represent it separately under different tabs of Genomic Variation, Gene Annotation, Functional Effects, Regulatory Variations, Strain Information, Drug Resistance, ncRNA and Genome Browser.(APPENDIX-IV)
2.	annoTB_upload.cgi	To extract the inserted variant file in the annoTB web-page, retrieve the corresponding information from the compiled

		datsheet and report Drug Resistant variations, Deleterious variations, Non-Synonymous variation, Synonymous variations, Regulatory Variations and the Novel variations that were not mapped on either of the databases. (APPENDIX-V)
3.	annoTB_submit.cgi	To save the novel variations submitted by the user on the server. (APPENDIX-VI)
4.	feedback_form.cgi	To retrieve the information fed by the user in the contact page and save it on the server. (APPENDIX-VII)

External modules used in the scripts include:

CGI.pm To provide a consistent Application Programming Interface for receiving user input and producing HTML output

DBI.pm Connecting to the database

List::MoreUtils qw/ uniq / To use the unique function (uniq) for an array

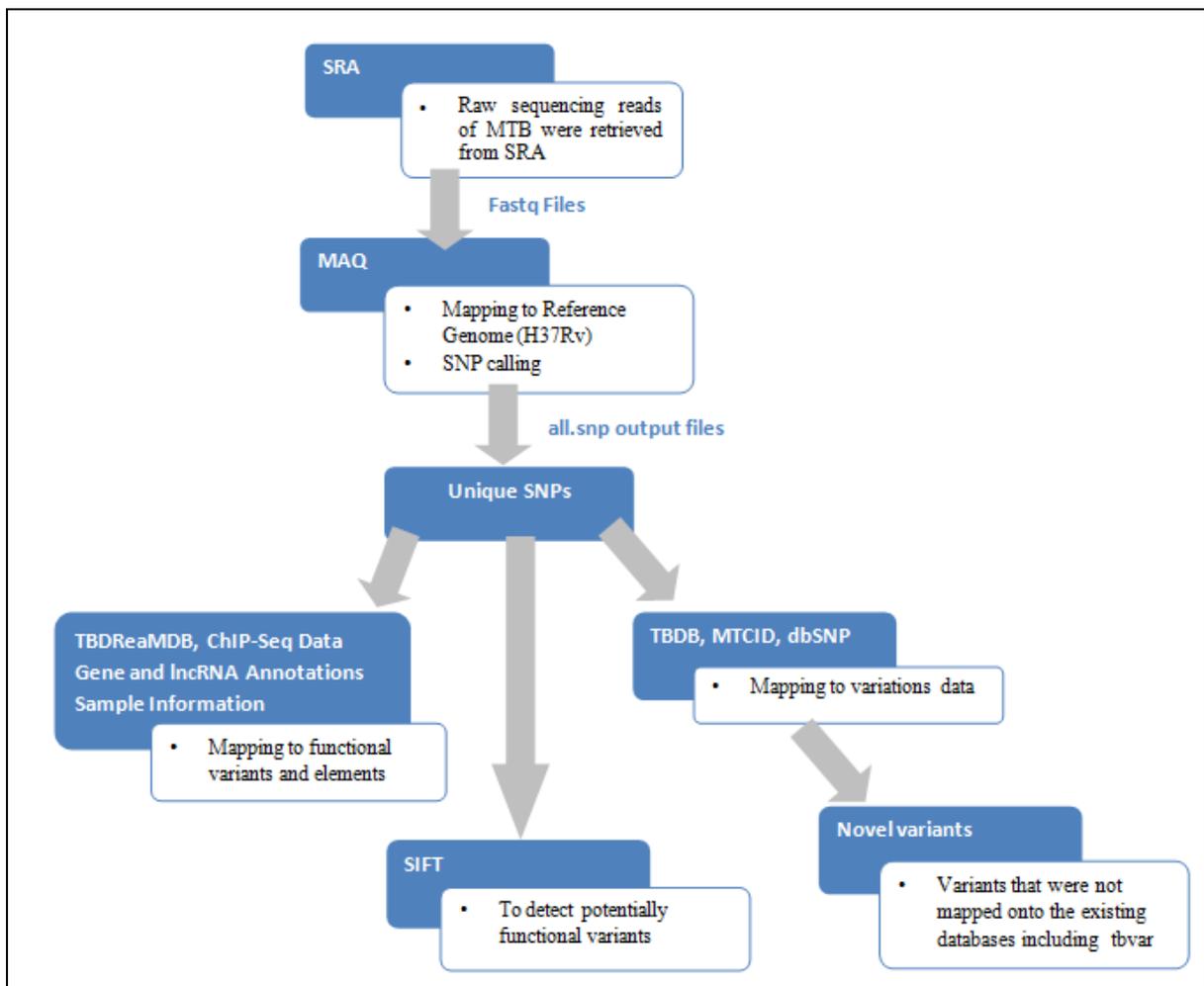


Fig. 22: Summary of the datasets and methodology used in creating the resource

4.13 Embedding JBrowse in the interface

JBrowse is a genome browser with a fully dynamic AJAX interface, being developed as the eventual successor to GBrowse. It is very fast and scales well to large datasets. It is fast to work with, performs smooth scrolling and zooming, enabling the user to explore the genome with unparalleled speed. JBrowse scales easily to multi-gigabase genomes. It can support various file formats that include GFF3, BED, FASTA, Wiggle, BigWig, BAM, and more. It can serve huge datasets from a single low-cost cloud instance.

1. Installing the prerequisites:

```
sudo apt-get install libpng-dev libgd2-noxpm-dev build-essential
```

2. Downloaded JBrowse from <http://jbrowse.org/jbrowse-1-9-7-maintenance-release/> onto the web server
3. Unpacked JBrowse and granted all permissions to the JBrowse directory

```
cd /var/www  
unzip JBrowse-1.9.7*.zip  
sudo chmod 777 JBrowse-1.9.7
```

- 5 To install all of JBrowse's (modest) prerequisites in the `jbrowse/` directory itself, the automated-setup script was run.

```
./setup.sh
```

- 6 Reference sequence was formatted for JBrowse using the following command:

```
bin/prepare-refseqs.pl -fasta docs/tutorial/data-files/mtb.fa
```

- 7 To import feature data into JBrowse, all the tracks from TBrowse (<http://tbrowse.osdd.net/>) and files with information of Genes and Variations were first converted in BED format. These flat-files were then imported using:

```
bin/flatfile-to-json.pl --out MTB/json/ --bed MTB/raw/*.bed --tracklabel
```

where * is the track name.

A shell script was written to import all the tracks with the same parameters as above wherein the above command was encoded for all the tracks.

- 8 After running the shell script, the browser could be accessed from localhost at the path:

```
jbrowse/index.html?data=MTB/json
```

- 9 To embed JBrowse within our web page, `iframe` tag was used within the `browser.html` script.

```
<iframe src="jbrowse/index.html?data=MTB/json" style="border: 1px solid black; width:900px; height:600px;"></iframe>
```

5. RESULTS

5.1 Data Compilation

The variants reported by MAQ were annotated using ANNOVAR, SIFT and *custom* scripts for mapping information regarding genes, regulatory peaks, ncRNAs, strain and drug resistance. The final output file was then loaded onto a database with respect to following column names:

Table V: Description of the columns that have been included while compiling the database

S. No.	Column Name	Description
1.	var_int_id	The variant position
2.	var_snp_id	ID's to link to the existing databases
3.	var_ref	Reference allele at the variant position
4.	var_alt	Alternate allele at the variant position
5.	var_loci	Whether the variant lies in the exonic region, Upstream or Downstream or Intergenic
6.	var_type	Whether the variation is Synonymous or Non-Synonymous
7.	var_freq	Frequency of finding a particular variation within the entire dataset. It is equal to the number of samples containing a particular variation.
8.	gene_int_id	RvID of the corresponding gene within which the variant lies
9.	gene_name	Name of the corresponding gene within which the variant lies
10.	gene_start	Genomic position from where the corresponding gene starts
11.	gene_stop	Genomic position from where the corresponding gene stops
12.	gene_orientation	“+”: for forward orientation, “-”: for reverse orientation
13.	var_aa_mutation	<reference amino acid><variant position><alternate allele>
14.	var_func_effect	Whether the variation is tolerated or deleterious
15.	var_sift_score	SIFT score corresponding to the variations which lies within 0-1
16.	var_TF	RvID or the gene name of the Transcription factor within which the variant position lies
17.	var_target	RvID or the gene name of the target of the corresponding TF
18.	reg_start	genomic position for the start of the regulatory peak
19.	reg_stop	Genomic position for the stop of the regulatory peak
20.	strain_sample	Accession Number of the corresponding sample
21.	strain_exp	Accession Number of the corresponding sample experiment
22.	strain_ref	Accession Number of the corresponding sample
23.	ncrna_start	genomic position for the start of ncRNA within which it falls
24.	ncrna_stop	genomic position for the stop of ncRNA within which it falls
25.	ncrna_name	Name or ID of the ncRNA within which it falls
26.	ncrna_strand	“+”: for forward strand, “-”: for reverse strand
27.	ncrna_product	Product description of the corresponding ncRNA
28.	gene_anno	Annotation of the gene within which the variation falls

5.2 Database statistics

Our database hosts over 29,472 unique single nucleotide variants from 469 unique sequenced strains of *M. tuberculosis*. Of the total, 7,856 variants could be mapped to other known variations in *M. tuberculosis* retrieved from dbSNP, MTCID & TBDB, suggesting that 21,616 variants are novel and reported for the first time in this report. The overlap of variations within tbvar with each of the resources is summarized in Figure 23.

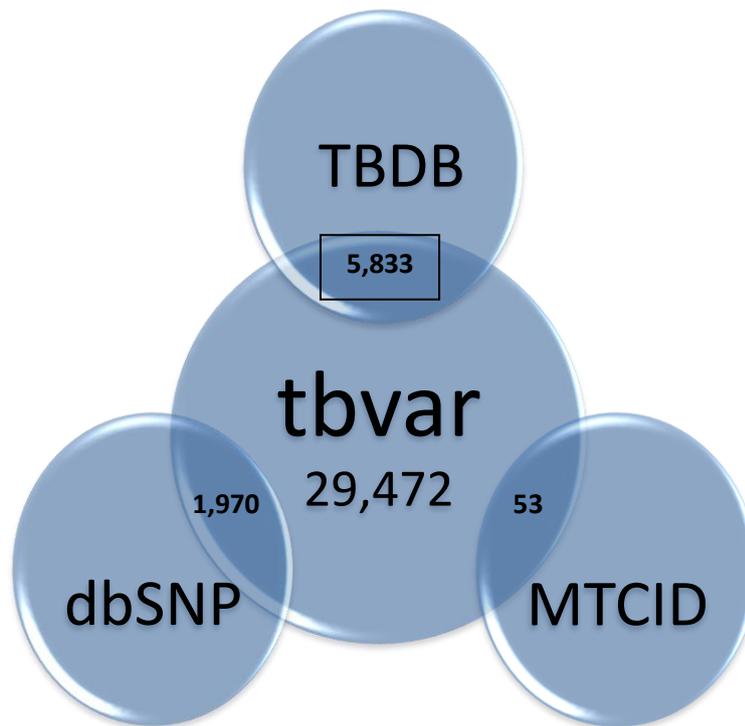


Fig. 23: Comparison of the variations in *M. tuberculosis* with respect to other variation resources.

From the entire dataset, 26,083 variations were found to lie within genes and 16,209 variations were reported as non-synonymous. In a total of 2,407 genes 5,394 variations were predicted as deleterious by SIFT. Apart from these a total of 9,446 variations were sense mutations and 398 confer stop codon mutation to the genes. A total of 38 mutations led to loss of a stop codon within gene and 7,873 variations mapped to regulatory regions annotated as per ChIP-seq dataset obtained from (<http://genome.tdb.org/annotation/genome/tdb/RegulatoryNetwork.html>). A comprehensive representation of the data is show in Figure 24.

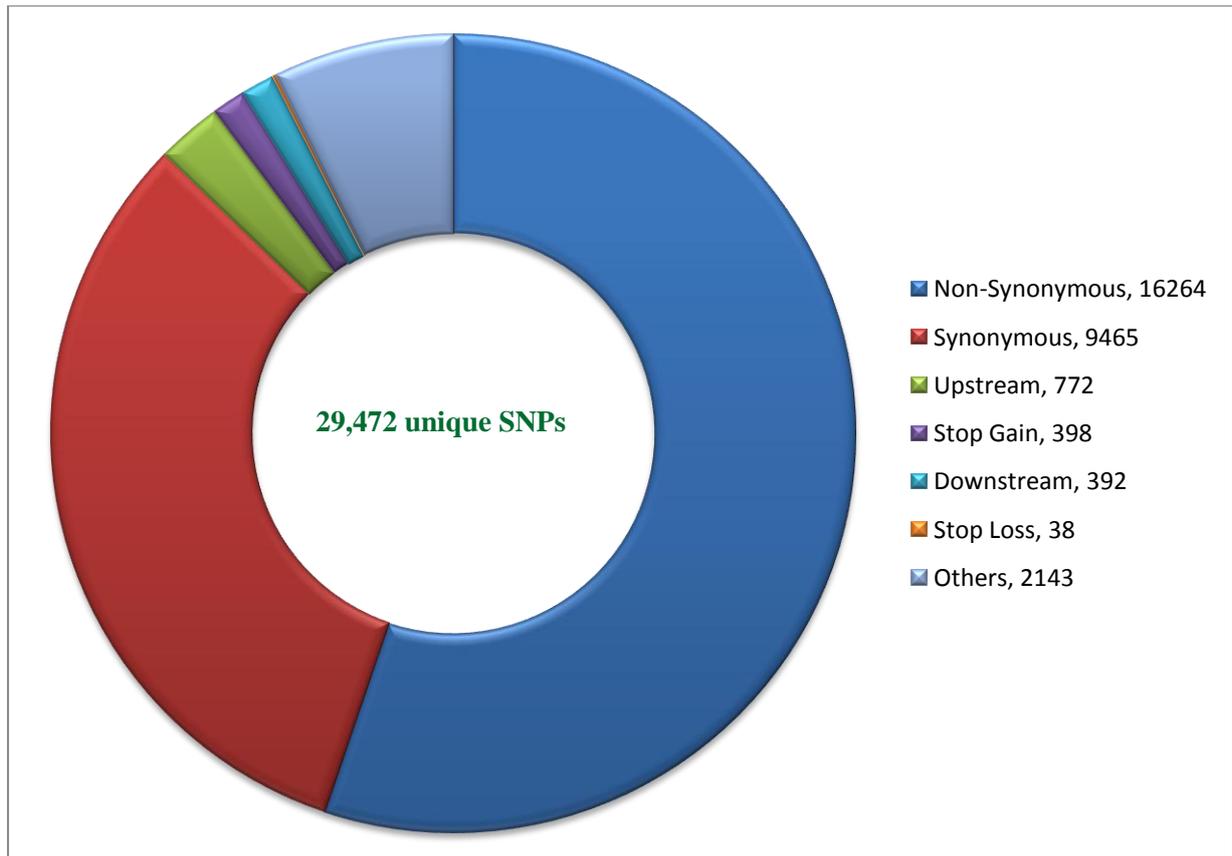


Fig. 24: Graphical representation showing distribution of SNPs in various loci of the *M. tuberculosis* genome

5.3 Genomic variations tend to saturation

We analyzed the repertoire of genomic variations encoded by the 469 (samples) strains of *M. tuberculosis* to evaluate whether this forms a near comprehensive set of genetic variations encoded by the pathogen. The percentage of the total repertoire contributed by randomly picked sets of genomes was evaluated.

1. Each bin of 5 samples was randomly chosen for 1000 iterations
2. Number of novel SNVs were identified for each bin
3. This number for each bin was averaged and a box plot was plotted

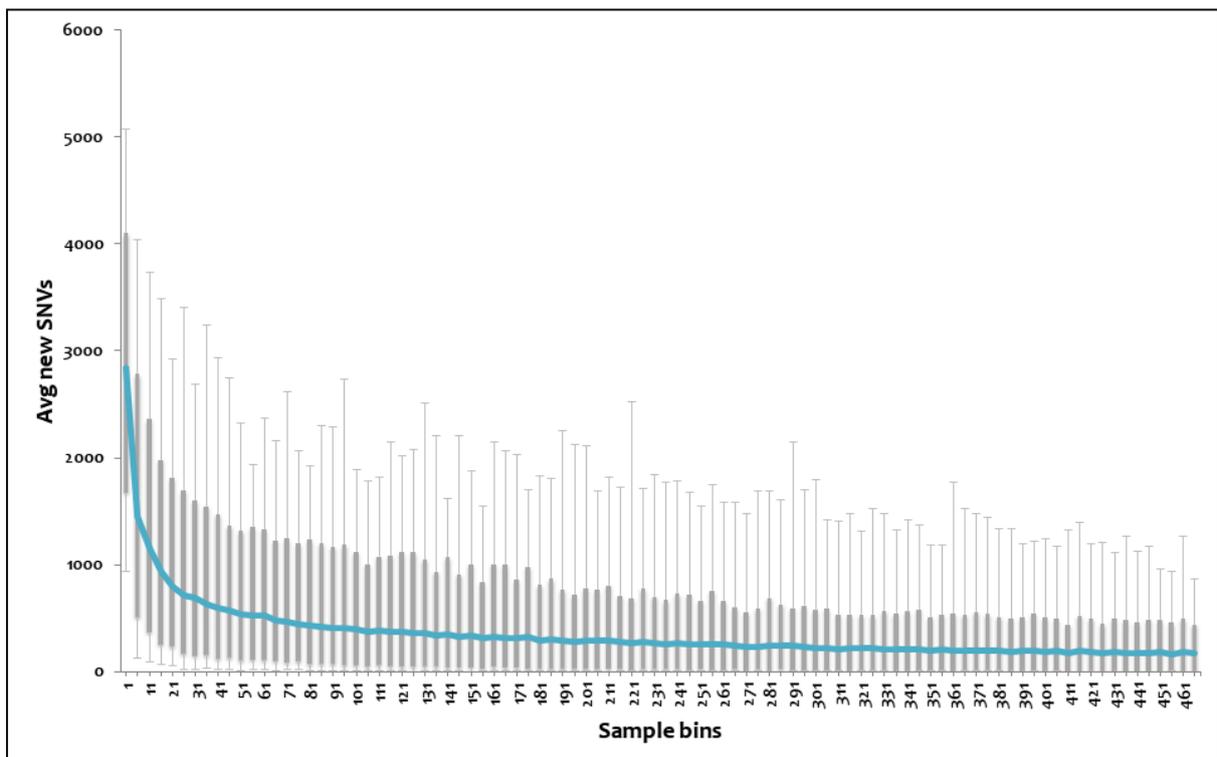


Fig. 25: Variations plotted across subset of the genomes. The 95 percentile and 5 percentile form upper and lower boxes, while upper and lower error bars indicate maximum and minimum. The blue line passing through the boxes indicates average per bin of samples.

It was found that the variations tend to saturate close to 195 genomes i.e. after 195 genomes not more than a fixed number of novel variations would come in any 10 samples taken at random. We suggest that tbvar thus encompasses majority of the common variants encoded by the pathogen, thus providing a comprehensive resource and starting point towards understanding the pathogen diversity and evaluation of variations for therapeutic and drug discovery applications.

5.4 Database features and navigation

5.4.1 Home Page

The database has a user-friendly interface. Homepage of the database introduces the database and its purpose to the users. It also gives a brief overview of the data that the database holds. It directs the user to different search and browse options. The search option is quite simple, where a user can search using a gene ID or a set of genomic positions and is immediately presented with a list of variants, which satisfy the condition. The search supports gene names, Tuberculist RvIDs and GenBank IDs. The 'Eureka' button returns results for the input search term in various tabs below the explore form. Search term in any of the above mentioned form takes user to a separate result page which is further divided into different sections in form of tabs. The result page interface is hyperlinked so as to provide the user to dynamic compilations of information, pertaining, for example a strain, a gene or a set of properties, say drug resistance. The homepage also features a browse interface which allows users to quickly browse for relevant information quickly. To aid the user on the genomic location of the variants under question, the resource also features a browser tab displaying the variations on a genome browser interface exported from TBrowse(Bhardwaj *et. al.*, 2009).

tbvar | Mycobacterium tuberculosis variome resource

annoTB | MANUAL | CONTACT

About tbvar

Mycobacterium tuberculosis is the causative organism of tuberculosis, a disease with high morbidity and mortality, especially in the developing world. The genetic variability between clinical isolates of this pathogen has been poorly understood. Recent years has seen the re-sequencing of a large number of clinical isolates for *Mycobacterium tuberculosis* from around the world. The availability of genomic data of multiple isolates in public domain offers a unique opportunity towards understanding the variome of the organism and the functional consequences of the variations. This necessitates systematic curation and analysis of datasets available in public domain.

In this report, we have re-analyzed data sets corresponding to over 400 isolates of Mtb available in public domain to reveal a comprehensive variome of Mtb comprising of over 29,000 single nucleotide variations, which has been deposited into a database (tbvar). Using a systematic computational pipeline, we have annotated potential functional variants and drug-resistance associated variants. Apart from a user-friendly interface, the database has a novel option to annotate variants from clinical re-sequencing of Mtb. To the best of our knowledge tbvar is the largest and most comprehensive genome variation resources for *Mycobacterium tuberculosis*.

Search for variations

Search: Variant Location, Range,

Database can be accessed by:

Browsing Variant Location
Example: 1417919 | 3037367 | 4222628

Browsing Genes
Example: katG | pncA | gyrA

Browsing RvID
Example: Rv1059 | Rv1083c | Rv3693

Browsing Genome Position Range
Example: 10000-15000 | 30000-35000 | 80000-85000

Browse the data

[tbvar Genome Browser](#)

Statistics

dbSNP: 1913, TBDBiv: 162, MTCID: 208

dbSNP only: 21635, TBDBiv only: 40, MTCID only: 208

dbSNP & TBDBiv: 3, TBDBiv & MTCID: 2, dbSNP & MTCID: 0, All three: 0

dbSNP & TBDBiv & MTCID: 0

dbSNP & MTCID: 0, TBDBiv & MTCID: 0, dbSNP & TBDBiv: 0

dbSNP & TBDBiv & MTCID: 0

Citation:
[Joshi et al (2013): tbvar: A comprehensive genome variation resource for *Mycobacterium tuberculosis*]

Funding and Support
We acknowledge the funding from the Open Source Drug Discovery Initiative/ CSIR for the programme. We also acknowledge the availability of communication and compute resources from NKN for the project.

Copyright©2013 CSIR-Institute of Genomics and Integrative Biology

Fig. 26: Home Page of the web-interface for tbvar

5.4.2 tbvar

The search form takes the user to a different page that hosts the complete information segregated under different tabs. Information regarding each tab can be obtained by hovering the mouse over the link. Each of the section displayed in the form of tab is highlighted in Figure 27. The interface also allows the user to shift through this list by using a set of tabs to filter out synonymous/non-synonymous, deleterious or variations in regulatory regions.

The screenshot shows the tbvar website interface. At the top, there is a navigation menu with links for HOME, annoTB, MANUAL, and CONTACT. Below the menu is a search box with a search button and a search box containing the text "Search Variant Location RefSeq, GI Eukarya". The main content area is divided into several sections, each with a tab: Genomic Variations, Gene Annotation, Functional Effects, Regulatory Variations, Strain Information, Drug Resistance, and Genome Browser. A table of genomic variations is displayed below the tabs, with columns for Gene ID, Position, Ref Allele, Alt Allele, Location, Type, Variant Count, Frequency Percentage, and External Link. The table contains 15 rows of data, with the first row highlighted in blue. The table is also filtered by the "Genomic Variations" tab.

Gene ID	Position	Ref Allele	Alt Allele	Location	Type	Variant Count	Frequency Percentage	External Link
Rv1900c	2154532	G	A	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2154724	C	A	exonic	non-synonymous SNV	175	37.31	TBDB, MTCD
Rv1900c	2154730	T	G	exonic	non-synonymous SNV	2	0.43	
Rv1900c	2154871	T	C	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2154902	A	G	exonic	synonymous SNV	2	0.43	
Rv1900c	2154988	G	T	exonic	non-synonymous SNV	1	0.21	4BSNP
Rv1900c	2155148	G	A	exonic	synonymous SNV	3	0.64	
Rv1900c	2155167	G	T	exonic	non-synonymous SNV	4	0.85	MTCD
Rv1900c	2155168	C	G	exonic	non-synonymous SNV	52	11.09	TBDB, MTCD
Rv1900c	2155168	C	T	exonic	non-synonymous SNV	6	1.28	TBDB, MTCD
Rv1900c	2155304	T	G	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2155343	T	C	exonic	non-synonymous SNV	2	0.43	
Rv1900c	2155412	C	T	exonic	non-synonymous SNV	5	1.07	
Rv1900c	2155503	G	A	exonic	synonymous SNV	3	0.64	TBDB
Rv1900c	2155726	A	G	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2155732	T	G	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2155751	C	T	exonic	non-synonymous SNV	1	0.21	
Rv1900c	2155783	G	A	exonic	non-synonymous SNV	1	0.21	MTCD
Rv1900c	2155832	C	T	exonic	non-synonymous SNV	1	0.21	

Fig. 27: Screenshot showing result table and information about each section of the database

The resource is hyperlinked so as to provide the user dynamic compilations of information, pertaining, for example a strain, a gene or a set of properties, say drug resistance. It is also interlinked to various other primary resources for gene information and sources of raw data. This includes Tuberculist and TBdb for gene information, Uniprot for protein annotations and NCBI SRA for raw datasets and TBrowse for genome centered annotations. The variants in the database are also available as a track shared in TBrowse. Variant datasets and annotations have also been made available for download to aid computational biologists with a ready set of formatted data sets for analyses.

The output page for database query is divided into different sections which can be explored using different tabs at the top of result table. Different sections show different biological and other information related to query.

Genomic variations:

This section gives information on the genomic position of the variations along with their genomic loci (e.g. whether they lie in genic part of the inter-genic region of the genome). This section also mentions the count and frequency of occurrence of the particular variation within the population of samples chosen for building database. Finally, this section also links out to other various databases for the variants found in other similar databases.

Gene Id	Position	Ref Allele	Alt Allele	Location	Type	Variant Count	Frequency Percentage	External Link
Rv1908c	2154332	G	A	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2154724	C	A	exonic	nonsynonymous SNV	175	37.31	TBDB:MTCID
Rv1908c	2154730	T	G	exonic	nonsynonymous SNV	2	0.43	
Rv1908c	2154871	T	C	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2154902	A	G	exonic	synonymous SNV	2	0.43	
Rv1908c	2154980	G	T	exonic	nonsynonymous SNV	1	0.21	dbSNP
Rv1908c	2155140	G	A	exonic	synonymous SNV	3	0.64	
Rv1908c	2155167	G	T	exonic	nonsynonymous SNV	4	0.85	MTCID
Rv1908c	2155168	C	G	exonic	nonsynonymous SNV	52	11.09	TBDB:MTCID
Rv1908c	2155168	C	T	exonic	nonsynonymous SNV	6	1.28	TBDB:MTCID
Rv1908c	2155301	T	G	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2155343	T	C	exonic	nonsynonymous SNV	2	0.43	
Rv1908c	2155412	C	T	exonic	nonsynonymous SNV	5	1.07	
Rv1908c	2155503	G	A	exonic	synonymous SNV	3	0.64	TBDB
Rv1908c	2155726	A	G	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2155732	T	G	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2155751	C	T	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2155783	G	A	exonic	nonsynonymous SNV	1	0.21	MTCID
Rv1908c	2155832	C	T	exonic	nonsynonymous SNV	1	0.21	
Rv1908c	2155945	T	G	exonic	nonsynonymous SNV	2	0.43	
Rv1908c	2155964	G	A	exonic	stopgain SNV	1	0.21	

Fig. 28: Information provided under ‘Genomic Variations’ tab

Gene annotation:

This section shows all the information pertaining to a gene harboring variations. It shows gene name and its ID. This tab also gives annotation for the respective gene. Genomic coordinates of genes and its orientation are also presented in this section.

Gene Id	Gene Name	Gene Annotation	Start	Stop	Orientation
Rv1908c	katG	Catalase-peroxidase-peroxytrinitrate T KatG	2153888	2156111	-

Fig. 29: Information provided under ‘Gene Annotation’ tab

Functional effects:

This section shows SIFT prediction and SIFT score for non-synonymous variations. SIFT predicts whether a genomic variation has any functional consequence on the corresponding protein. Here we consider a change in protein structure as predicted by SIFT to be DELETERIOUS, while no change in the protein structure is considered as TOLERATED. SIFT score is a major parameter in prediction and is provided in this tab.

Position	Amino Acid Mutation	Functional Effect	SIFT Score
2154532	S527L	DELETERIOUS	0.02
2154724	R463L	TOLERATED	1
2154730	Q461P	DELETERIOUS	0
2154871	K414R	DELETERIOUS	0
2154902	L404L	Not Scored	Not Scored
2154989	L379M	DELETERIOUS	0.03
2155140	T324T	Not Scored	Not Scored
2155167	S319R	DELETERIOUS	0
2155168	S315T	DELETERIOUS	0
2155168	S319N	DELETERIOUS	0
2155301	T271P	DELETERIOUS	0
2155343	M257V	DELETERIOUS	0
2155412	G234R	DELETERIOUS	0
2155503	T203T	Not Scored	Not Scored
2155726	F129S	DELETERIOUS	0
2155732	Q127P	DELETERIOUS	0
2155751	G121S	DELETERIOUS	0
2155783	A110V	DELETERIOUS	0
2155832	D94N	DELETERIOUS	0
2155945	D56A	DELETERIOUS	0.01
2155964	Q50K	Not Scored	Not Scored

Fig. 30: Information provided under 'Functional Effects' tab

Regulatory variations:

Variations lying in the regulatory elements on MTB genome as found through Transcription factor (TF) ChIP-seq of 50 transcription factors are reported in this section. The targets predicted against these TFs are also reported in this section along with their genomic positions for start and stop. In case the variation doesn't lie within the range of genomic coordinates of the peak, a message of "No Regulatory Effect is shown".

Position	Regulator	Target	Start	Stop
2154532	Rv3133c	Rv1908c	2154420	2154639
2154532	Rv3133c	Rv1908c	2154421	2154645
2154724	Rv3133c	Rv1908c	2154711	2154947
2154724	Rv2089	Rv1908c	2154549	2154888
2154730	Rv3133c	Rv1908c	2154711	2154947
2154730	Rv2089	Rv1908c	2154549	2154888
2154871	Rv3133c	Rv1908c	2154711	2154947
2154871	Rv2089	Rv1908c	2154549	2154888
2154871	Rv3133c	Rv1908c	2154758	2154928
2154902	Rv3133c	Rv1908c	2154711	2154947
2154902	Rv3133c	Rv1908c	2154758	2154928
2154989	No Regulatory Effect			
2155140	No Regulatory Effect			
2155167	No Regulatory Effect			
2155168	No Regulatory Effect			
2155301	No Regulatory Effect			
2155343	No Regulatory Effect			
2155412	No Regulatory Effect			
2155503	No Regulatory Effect			
2155726	No Regulatory Effect			
2155732	No Regulatory Effect			
2155751	No Regulatory Effect			

Fig. 31: Information provided under 'Regulatory Variations' tab

Strain information:

This section reports information on the samples and corresponding experiment and study from which the variations were derived. In case of multiple samples, experiments or studies for a single variation, all of them are listed one after the other in successive rows.

tbvar Mycobacterium tuberculosis variome resource

HOME annoTB MANUAL CONTACT

Search: Variant Location, Range, Gene or Rvid Eureka

Genomic Variations Gene Annotation Functional Effects Regulatory Variations **Strain Information** Drug Resistance Genome Browser

Position	Sample	Experiment	Reference
2154532	ERS019751	ERX013857	ERP000436
2154724	ERS019735	ERX013859	ERP000436
2154724	ERS019739	ERX013859	ERP000436
2154724	ERS019741	ERX013859	ERP000436
2154724	ERS019754	ERX013857	ERP000436
2154724	ERS019755	ERX013857	ERP000436
2154724	ERS019757	ERX013857	ERP000436
2154724	ERS019760	ERX013860	ERP000436
2154724	ERS019761	ERX013860	ERP000436
2154724	ERS019762	ERX013860	ERP000436
2154724	ERS019766	ERX013860	ERP000436
2154724	ERS019767	ERX013860	ERP000436
2154724	ERS019768	ERX013860	ERP000436
2154724	ERS019773	ERX013856	ERP000436
2154724	ERS019778	ERX013856	ERP000436
2154724	ERS019779	ERX013856	ERP000436
2154724	ERS019785	ERX013858	ERP000436
2154724	ERS020002	ERX014930	ERP000436
2154724	ERS020005	ERX014930	ERP000436
2154724	ERS020012	ERX014928	ERP000436
2154724	ERS020018	ERX014928	ERP000436
2154724	ERS020019	ERX014928	ERP000436

Fig. 32: Information provided under 'Strain Information' tab

Drug resistance:

Variations known to be annotated as drug resistant are reported in this section of the database. Variations having resistance to known anti-tb drug along with the antibiotic and corresponding resistant gene information are reported under this tab. Peer reviewed reference from which the information was derived is also reported. This tab appears only if the query has a drug resistant mutant associated with it.

tbvar Mycobacterium tuberculosis variome resource

HOME annoTB MANUAL CONTACT

Search: Variant Location, Range, Gene or Rvid Eureka

Genomic Variations Gene Annotation Functional Effects Regulatory Variations Strain Information **Drug Resistance** Genome Browser

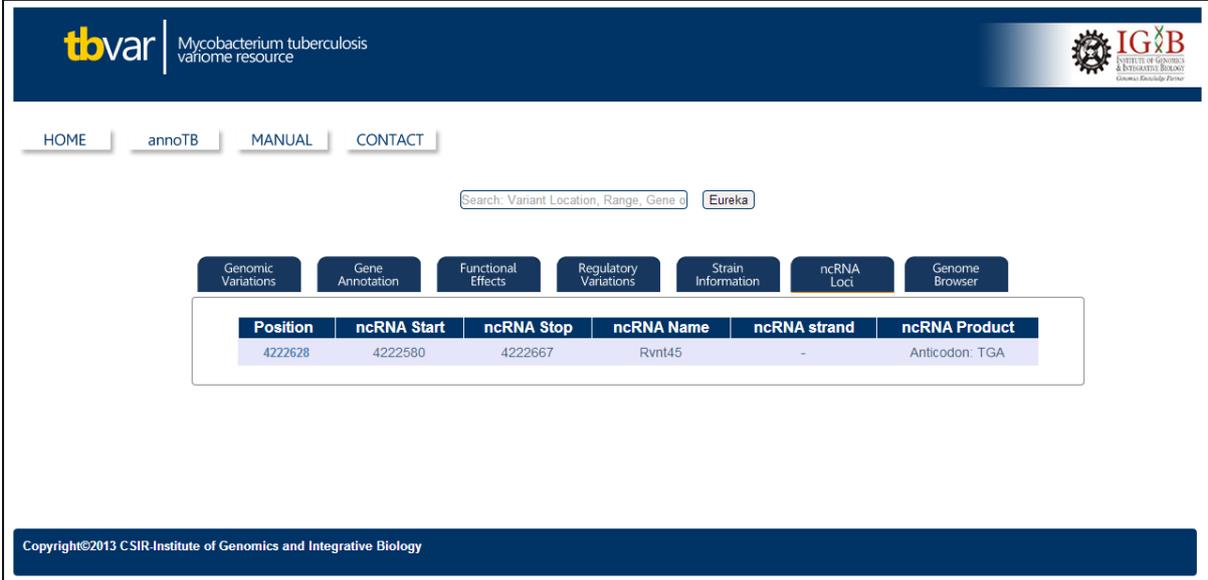
Position	Resistant Drug	Resistant Gene	Reference
2155167	INH	katG	Lipin MY CMI 2007
2155168	INH	katG	Musser JM JID 1996
2155412	INH	katG	Chan RCY JAC 2007
2155732	INH	katG	Chan RCY JAC 2007
2155751	INH	katG	Gagneux S PLOS Path 2006
2155783	INH	katG	Wei CJ AAC 2003

Copyright©2013 CSIR

Fig. 33: Information provided under 'Drug Resistance' tab

ncRNA Loci:

Variations falling in the non-coding RNA regions of the genome are reported in this tab. This is also an optional tab which appears only if any variation in the query lies in the non-coding region of the genome. It reports genomic coordinates of the nc-RNA, its orientation and the product of this ncRNA.



The screenshot shows the tbvar Mycobacterium tuberculosis variome resource interface. At the top, there is a search bar with the text "Search: Variant Location, Range, Gene o" and a button labeled "Eureka". Below the search bar are several navigation tabs: "Genomic Variations", "Gene Annotation", "Functional Effects", "Regulatory Variations", "Strain Information", "ncRNA Loci", and "Genome Browser". The "ncRNA Loci" tab is selected, and a table of data is displayed below it.

Position	ncRNA Start	ncRNA Stop	ncRNA Name	ncRNA strand	ncRNA Product
4222628	4222580	4222667	Rvnt45	-	Anticodon: TGA

At the bottom of the interface, there is a copyright notice: "Copyright©2013 CSIR-Institute of Genomics and Integrative Biology".

Fig. 34: Information provided under 'ncRNA loci' tab

Genome Browser:

A genome browser hosting tracks from TBrowse and other relevant variation information and region surrounding the region of interest is shown in this section. Users can navigate through the genome by clicking and dragging the browser area. New tracks of interest can be added to the browser by a simple drag and drop from the vertical panel on the left.

Users can also upload their own tracks of interest to show up in the browser by choosing:

In case of local files:

File -> Open -> Select Files

Or

Drag and Drop the files into the text box

In case of remote files:

Paste remote URLs.

Users can upload files in one of these file formats (GFF, BigWig, BAM & BAI). Configuration of the information to show in browser is customizable. Users have an option to either open the file or directly add it as a track.



Fig. 35: Information provided under ‘Genome Browser’ tab

5.4.3 Application of tbvar: annoTB

One of the ready applications of a comprehensive resource like tbvar would be for annotation of variants from re-sequencing of isolates, including in clinical settings. tbvar provides an annotation feature ‘annoTB’, where users can input custom variant list from re-sequencing data, and get annotations for them. In the present report we use this feature as a proof of concept to annotate variations from a clinical isolate.

This is novel feature of tbvar, where clinicians can simply upload MTB variant list in a simple file format which is used by a parser to map those variations onto variations housed in ‘tbvar’. Since tbvar is a comprehensive compendium of known genomic variations in MTB, most of the variations in clinical sample genome get annotated. Rest of the variations that don’t map onto the in-house variant list are considered to be novel, ones that have never been reported. For such variations tbvar provides an option to submit them to the IGIB server.

Input: annoTB accepts input in the form of an SNP file. An example SNP file can be viewed by clicking on ‘Load Example’ button below the text area. annoTB only accepts variant position, reference allele and alternate allele, in tab separated format. Ambiguous bases are not allowed for annotation and hence discarded.

Pos.	Ref.	Alt.
30944	C	T
30954	A	T
31468	T	C
31468	T	C
34063	C	T
35063	C	A
7585	G	C

tbvar Mycobacterium tuberculosis variome resource

HOME annoTB MANUAL CONTACT

Batch query: **Simple text input for variation**

30944 C T
30954 A T
31468 T C
31468 T C
34063 C T
35063 C A
7585 G C
81649 C G
8428 G T
9557 G C
9567 T C
1417019 C T
3037367 C T
4222628 C T
2134532 G A
2164532 C A

Add/Delete
Submit Query Reset → **Example input**
Reset button
Upload SNPs

REPORT

Uploaded Variants: 19
Mapped Variants: 14
Novel Variants: 5
Drug Resistant Variants: 1

Drug Resistant Variants

Variant Position	Ref. Allele	Alt. Allele	Variant Count	Gene	Effect
7585	G	C	1	rpoB	Drug Resistance

Synonymous Variants

Variant Position	Ref. Allele	Alt. Allele	Variant Count	Gene	Effect
30944	C	T	1	IS6110	Synonymous

Non-Synonymous Variants

Variant Position	Ref. Allele	Alt. Allele	Variant Count	Gene	Effect
30954	A	T	1	IS6110	Non-Synonymous

Regulatory Variants

Variant Position	Ref. Allele	Alt. Allele	Variant Count	Gene	Effect
3037367	C	T	1	IS6110	Regulatory

Novel Variants

Variant Position	Ref. Allele	Alt. Allele	Variant Count	Gene	Effect
4222628	C	T	1	IS6110	Novel

Report summary

Fig. 36: Description of the annoTB web-page

Output:

A report showing variation annotation and corresponding information is available for users to analyze. The report gives a summary of annotated variations and also shows the variations known to confer drug resistance. Different panels below report summary give information on annotation of individual variation section-wise.

Report Summary lists the certain numbers of interest for the user. It tells how many variants were uploaded, amongst which how many got mapped onto the in-house variant dataset. Out of the mapped dataset, numbers of those that have corresponding drug resistance annotations, are synonymous or non-synonymous or reported to have regulatory effect are also given. Number of variations that don't map to the tbvar dataset (novel variations) is also reported here. Apart from this the report summary also informs the resistant drugs for the entire input data, along with the corresponding variations beneath them.

REPORT

Uploaded Variants: 19
Mapped Variants: 14
Novel Variants: 5
Drug Resistant Variants: 1

Synonymous Variants: 2
Non-Synonymous Variants: 10
Regulatory Variations: 6
Ambiguous bases removed: 0

Resistant Drugs:

FLQ
7585

Fig. 37: Information provided under annoTB Report Summary

Drug resistance panel lists the variations annotated to be drug resistant.

Drug Resistant Variations							
Variant Position	Ref Allele	Alt Allele	Variant Count	Gene	Type	Resistant Drug	
7585	G	C	75.74	gyrA	nonsynonymous SNV	FLQ	
781822	A	G	1.49	rpsL	nonsynonymous SNV	SM	
1417019	C	T	7.23	embR	nonsynonymous SNV	EMB	
2155168	C	G	12.13	katG	nonsynonymous SNV	INH	
2155168	C	T	12.13	katG	nonsynonymous SNV	INH	
2517610	G	A	5.53	fabD	nonsynonymous SNV	INH	
4240671	C	T	8.72	embC	nonsynonymous SNV	EMB	
4241042	A	G	7.02	embC	nonsynonymous SNV	EMB	
4245969	C	T	7.23	embA	nonsynonymous SNV	EMB	
4247730	G	A	1.28	embB	nonsynonymous SNV	EMB	

Fig. 38: Information related to ‘Drug Resistant Variations’ provided under annoTB Report

Deleterious variations panel lists deleterious variations predicted by SIFT which exist in database.

Deleterious Variations						
Variant Position	Ref Allele	Alt Allele	Variant Count	Gene	SIFT Score	
3446	C	T	3.19	recP	0.03	
8452	C	T	7.23	gyrA	0.01	
11879	A	G	81.49	Rv000c	0.04	
13298	G	C	7.23	Rv0010c	0.03	
20544	G	C	1.28	rodA	0.03	
26347	C	G	7.23	Rv0021c	0	
65663	C	G	7.23	celA1	0	
84528	T	G	5.53	Rv0075	0	
98966	G	C	7.23	Rv0090	0	
189850	A	G	3.19	PE4	0	
234268	G	T	1.28	Rv0187	0.04	

Fig. 39: Showing the information related to ‘Deleterious Variations’ provided under annoTB Report

Non-synonymous and synonymous variations are listed in next two panels wherein the user can know which of his input variations are synonymous and which ones are non-synonymous.

Non-synonymous Variations						
Variant Position	Ref Allele	Alt Allele	Variant Count	Gene	Type	
7585	G	C	75.96	gyrA	nonsynonymous SNV	
8428	G	T	0.21	gyrA	nonsynonymous SNV	
30944	C	T	0.21	Rv0026	nonsynonymous SNV	
31468	T	C	0.21	Rv0027	nonsynonymous SNV	
1417019	C	T	6.81	embR	nonsynonymous SNV	
2154532	G	A	0.21	katG	nonsynonymous SNV	
2154730	T	G	0.43	katG	nonsynonymous SNV	
2155412	C	T	1.06	katG	nonsynonymous SNV	
2155783	G	A	0.21	katG	nonsynonymous SNV	
4222628	C	T	0.21	Rv3776	nonsynonymous SNV	

Synonymous Variations						
Variant Position	Ref Allele	Alt Allele	Variant Count	Gene	Type	
9557	G	C	0.21	gyrA	synonymous SNV	
81649	C	G	1.70	Rv0072	synonymous SNV	

Fig. 40: Information related to ‘Syn/Non-Syn Variations’ provided under annoTB Report

Regulatory variations in uploaded SNP file matching to those present in database are shown in regulatory variation panel.

Regulatory Variations								
Variant Position	Ref Allele	Alt Allele	Variant Count	Gene	Type	Regulator	Target	
1977	A	G	76.17			Rv0081	Rv0002	
1977	A	G	76.17			Rv0081	Rv0001	
1977	A	G	76.17			Rv0081	Rv0002	
1977	A	G	76.17			Rv0081	Rv0001	
3446	C	T	3.19	recF	nonsynonymous SNV	Rv0324	Rv0002	
3446	C	T	3.19	recF	nonsynonymous SNV	Rv0324	Rv0003	
4013	T	C	80.85	recF	nonsynonymous SNV	Rv3574	Rv0004	
4013	T	C	80.85	recF	nonsynonymous SNV	Rv3574	Rv0002	

Fig. 41: Information related to ‘Regulatory Variations’ provided under annoTB Report

Novel Variations: The last panel lists those variations not present in ‘tbvar’. Users have an option to submit these novel variations to the database.

Novel Variations			
Variant Position	Ref Allele	Alt Allele	
2369	A	G	
12697	C	S	
13304	G	C	
17088	G	C	
23627	C	A	
24721	A	R	
46426	C	G	
55308	G	K	
55540	A	R	

Fig. 42: Information related to ‘Novel Variations’ provided under annoTB Report

When users press the submit button, a form asking for information of the submitter and submission opens up. By submitting the form users submit the SNP file they uploaded to the server where manual curation is done and the data is included into the database.

Mycobacterium tuberculosis
variant resource

IGB
INTEGRATED GENOMICS
& BIOMATERIALS
GENOMICS RESEARCH CENTER

HOME
annoTB
MANUAL
CONTACT

Your Name:

Institution:

e-mail ID:

Strain:

Publication:

Reference Genome:

Geographic Location:

Minimum Depth Coverage:

Fig. 43: Submission form for submitting variant file that has been loaded by the user in annoTB

5.4.4 Help Manual

The MANUAL button links the web page to a detailed Help Manual which is provided to aid the user on the search and navigation options. The manual has been embedded in the webpage using iframe tags (<iframe></iframe>). The manual is also available for download from the Download Manual link.



Fig. 44: Screenshot of web-interface for the Reference Manual

5.4.5 Contact Page

The users have been given an option to contact us by filling the form that has been linked to the CONTACT button. Herein the user is asked to enter its name, e-mail id and comments regarding the database. Whatever the user fills in is fetched by the CGI script feedback_form.cgi with a click on the submit button. This information is saved as a file with the name of [user]_[id].txt in the folder named 'feedback' on the server.

tbvar | Mycobacterium tuberculosis
variome resource

IGB
INSTITUTE OF GENOMICS
& INTEGRATIVE BIOLOGY
Council of Scientific and Industrial Research

HOME annoTB MANUAL ABOUT US CONTACT

Your Name:

e-mail ID:

Your Comments:

Submit

Copyright©2013 CSIR Institute of Genomics and Integrative Biology

Fig. 44: Screenshot of web-interface for the Contact Us form

6. DISCUSSION

Present day concern is a number of interconnected issues that include high morbidity and mortality rate of tuberculosis, steady growth in the reported incidence of drug-resistant isolates of MTB to the principal anti-tuberculosis drugs, resequencing of large number of clinical isolates from around the world and lack of a comprehensive, well-curated and user-friendly database dedicated to SNP data. With the advent of Next Generation Sequencing Technologies, a significant amount of SNP data is being developed worldwide. Owing to the need of the hour with respect to MTBC analysis, in order to cater in the prevention and cure of tuberculosis, it is highly recommended that this data should not be computed and stored in local workstations, rather deposited onto the Sequence Read Archive created by NCBI, EBI and DDBJ. Deposition of the data to SRA would make it publically available, so that the analysis can be done at a larger scale, worldwide.

Five existing databases harbour information on polymorphisms within MTBC that include TBDB, MTCID, MGDD, dbSNP and TBDreaMDB. The most important and multi-functional amongst these is TBDB which contains 23,795 SNPs extracted from 25 MTBC genomes. While, TBDreaMDB is known to be the most complete repository for drug resistance mutations in MTBC till date, that features 1447 variations. A new database for MTBC is highly required that is frequently updated and contains explicit annotations corresponding to the mutations. It should include certain important fields like essentiality of the corresponding gene based on experimental evidence, source of that variation like the sample information, clinical associations like virulence, drug treatment etc., functional predictions for the variants and frequency data on SNP distribution, in addition to the custom fields that include position, gene annotation, nucleotide change, amino acid change, type of amino acid change (synonymous/non-synonymous) etc.

So the major objectives of this study were to analyze re-sequencing data sets from various laboratories in public domain, identify genomic variations amongst these datasets, annotate them and characterize them with functional consequences and associated drug resistance and finally provide a near-comprehensive annotated repertoire of common genomic variations. tbvar provides a much needed compendium of genomic variants and annotations for *Mycobacterium tuberculosis* and provides the first step towards accelerating genotype-phenotype correlations in the pathogen. It houses 29,472 unique single nucleotide variants from 469 sequenced strains of *M. tuberculosis*. The database also provides a user-friendly interface, closely integrated and interlinked with other major resources in the field. Of the total, 7,856 variants could be mapped to other known variations in *M. tuberculosis* retrieved from dbSNP, MTCID & TBDB, suggesting that 21,616 variants are novel and reported for the first time in this report. 26,083 variations were genic and 16,209 were found to be non-synonymous and 5,394 were found to be deleterious in a total of 2,407 genes as predicted by SIFT, while a total of 9,446 variations were sense mutations and number of variations conferring stop codon mutation to the gene was 398. A total of 38 mutations led to loss of a stop codon within gene. A total of 7,873 variations mapped to regulatory regions.

tbvar also provides a novel application 'annoTB', where users can input custom variant list from re-sequencing data, and get them annotated for potential drug-resistant variations, retrieve allele frequencies of variations, know the genomic variations in regulatory regions and also retrieve information on other strains which have the same variation.

7. CONCLUSION AND FUTURE PERSPECTIVE

NGS studies of MTBC clinical isolates are discovering thousands of SNPs. Studying the functional effects of these SNPs and their association with phylogenetic clades is required to become an increasing concern of the research portfolio. MTBC consist of a diverse population of strains, and this diversity is considered to be important while developing new tools and strategies to combat TB. A new, extended, and well-curated database is thus, necessary to accommodate these rapidly accumulating SNP data in a user-friendly and integrated format.

Treading to achieve this motive, an initiative was taken to design tbvar. The variome was deciphered by mapping all the high quality sequencing reads from the publically available samples (469 samples) out of embargo to the reference genome (H37Rv). Keeping into consideration the existing features as well as the missing features within the currently available databases, an explicit list of parameters was decided to provide annotations to the identified variome. tbvar, thus encompasses majority of the common variants encoded by the pathogen, providing a comprehensive resource and starting point towards understanding the pathogen diversity and evaluation of variations for therapeutic and drug discovery applications.

We foresee drastic improvements in the compendium of genomic variants with more genome scale data being available in public domain. We would also improve upon the variant annotations with availability of genome-scale epigenetic data sets from large consortia in public domain, consistently improving the functional annotation of variations.

8. REFERENCES

- Ajay, S. S., Parker, S. C.; Abaan, H. O.; Fajardo, K. V.; and Margulies, E. H. (2011). Accurate and comprehensive sequencing of personal genomes. *Genome Res* **21**(9): 1498-1505.
- Astier, Y.; Braha, O. and Bayley, H.; (2006). Toward single molecule DNA sequencing: direct identification of ribonucleoside and deoxyribonucleoside 5'-monophosphates by using an engineered protein nanopore equipped with a molecular adapter. *J Am Chem Soc* **128**(5): 1705-1710.
- Aubry, A.; Pan, X. S. ; Fisher, L. M. ; Jarlier V. ; and Cambau E. (2004). Mycobacterium tuberculosis DNA gyrase: interaction with quinolones and correlation with antimycobacterial drug activity. *Antimicrob Agents Chemother* **48**(4): 1281-1288.
- Bentley, D. R. (2006). Whole-genome re-sequencing. *Curr Opin Genet Dev* **16**(6): 545-552.
- Bhardwaj, A.; Bhartiya, D.; Kumar, N.; and Scaria, V. ; (2009). TBrowse: An integrative genomics map of Mycobacterium tuberculosis. *Tuberculosis* **89**(5): 386-387.
- Bharti, R.; Das, R.; Sharma P.; Katoch K.; and Bhattacharya A. (2012). MTCID: a database of genetic polymorphisms in clinical isolates of Mycobacterium tuberculosis. *Tuberculosis (Edinb)* **92**(2): 166-172.
- Blanchard, J. S. (1996). Molecular mechanisms of drug resistance in Mycobacterium tuberculosis. *Annu Rev Biochem* **65**: 215-239.
- Breitling, R. (2010). What is systems biology? *Frontiers in Physiology*. **1**(9)
- Brosch, R.; Gordon, S. V. ; Marmiesse, M.; Brodin, P. ; Buchrieser, C. ; Eiglmeier, K. ; Garnier, T.; Gutierrez, C. ; Hewinson, G.; Kremer, K.; Parsons, L. M.; Pym, A. S.; Samper, S.; Soolingen D. V.; and Cole, S. T. (2002). A new evolutionary scenario for the Mycobacterium tuberculosis complex. *Proc Natl Acad Sci U S A* **99**(6): 3684-3689.
- Cole, S. T.; Brosch, R.; Parkhill, J.; Garnier, T. ; Churcher, C.; Harris, D.; Gordon, S. V.; Eiglmeier, K.; Gas, S.; Barry, C. E. ; Tekaiia, F.; Badcock, K.; Basham, D.; Brown, D.; Chillingworth, T. ; Connor, R.; Davies, R.; Devlin, K.; Feltwell, T. ; Gentles, S. ; Hamlin, N.; Holroyd, S. ; Hornsby, T.; Jagels, K. ; Krogh, A.; McLean, J.; Moule, S.; Murphy, L.; Oliver, K. ; Osborne, J.; Quail, M. A.; Rajandream, M. A. ; Rogers, J. ; Rutter, S. ; Seeger, K. ; Skelton, J.; Squares, R.; Squares, S.; Sulston, J. E. ; Taylor, K.; Whitehead S.; and Barrell B. G.; (1998). Deciphering the biology of Mycobacterium tuberculosis from the complete genome sequence. *Nature* **393**(6685): 537-544.
- Comas, I.; Chakravarti, J. ; Small, P. M. ; Galagan, J.; Niemann, S. ; Kremer, K. ; Ernst J. D. ; and Gagneux S. (2010). Human T cell epitopes of Mycobacterium tuberculosis are evolutionarily hyperconserved. *Nat Genet* **42**(6): 498-503.
- Comas, I.; and Gagneux S. (2009). The past and future of tuberculosis research. *PLoS Pathog* **5**(10): e1000600.
- Coscolla, M. and Gagneux, S. (2010). Does M. tuberculosis genomic diversity explain disease diversity? *Drug Discov Today Dis Mech* **7**(1): e43-e59.

- De Rossi, E.; Ainsa, J. A. and Riccardi, G. (2006). Role of mycobacterial efflux transporters in drug resistance: an unresolved question. *FEMS Microbiol Rev* **30**(1): 36-52.
- Ewing, B. and Green, P. (1998). Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* **8**(3): 186-194.
- Ewing, B.; Hillier, L. ; Wendl, M. C. and Green, P. (1998). Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res* **8**(3): 175-185.
- Fleischmann, R. D.; Alland, D. ; Eisen, J. A.; Carpenter, L.; White, O.; Peterson, J. ; DeBoy, R.; Dodson, R.; Gwinn, M. ; Haft, D. ; Hickey, E.; Kolonay, J. F. ; Nelson, W. C.; Umayam, L. A. ; Ermolaeva, M.; Salzberg, S. L.; Delcher, A. ; Utterback, T. ; Weidman, J. ; Khouri, H. ; Gill, J. ; Mikula, A. ; Bishai, W. ; Jacobs Jr, W. R.; Venter, J. C. ; and Fraser, C. M. (2002). Whole-genome comparison of *Mycobacterium tuberculosis* clinical and laboratory strains. *J Bacteriol* **184**(19): 5479-5490.
- Ford, C.; Yusim, K.; Ioerger, T. ; Feng, S. ; Chase, M. ; Greene, M. ; Korber, B. and Fortune, S. (2012). *Mycobacterium tuberculosis* – Heterogeneity revealed through whole genome sequencing. *Tuberculosis* **92**(3): 194-201.
- Ford CB, L. P.; Chase, M.R.; Shah, R.R.; Iartchouk, O; Galagan, J; Mohaideen, N; Ioerger, T.R.; Sacchettini, J.C.; Lipsitch, M.; Flynn, J.L.; Fortune, S.M. (2011). Use of whole genome sequencing to estimate the mutation rate of *Mycobacterium tuberculosis* during latent infection.pdf. *Nature genetics* **43**: 389-499.
- Fumagalli, D. and Sotiriou, C. (2013). Promises and Challenges of the Next Generation Sequencing in Breast Cancer Drug Development. <http://www.icact.fr/icours/icact/d2r1ps1p1/d2r1ps1p1.html>
- Gagneux, S. and Small, P. M. (2007). Global phylogeography of *Mycobacterium tuberculosis* and implications for tuberculosis product development. *Lancet Infect Dis* **7**(5): 328-337.
- Gillespie, S. H. (2002). Evolution of drug resistance in *Mycobacterium tuberculosis*: clinical and molecular perspective. *Antimicrob Agents Chemother* **46**(2): 267-274.
- Greenleaf, W. J. and Block, S. M. (2006). Single-molecule, motion-based DNA sequencing using RNA polymerase. *Science* **313**(5788): 801.
- Illumina. (2011). Quality Scores for Next Generation Sequencing. Technical note: Sequencing. http://res.illumina.com/documents/products/technotes/technote_q-scores.pdf
- Ioerger, T. R.; Feng, Y. ; Ganesula, K. ; Chen, X.; Dobos, K. M.; Fortune, S. ; Jacobs, W. R. ; Mizrahi, V.; Parish, T.; Rubin, E.; Sasseti C. ; and Sacchettini, J. C. (2010). Variation among Genome Sequences of H37Rv Strains of *Mycobacterium tuberculosis* from Multiple Laboratories. *Journal of Bacteriology* **192**(14): 3645-3653.
- Jarlier, V. and Nikaido, H. (1994). Mycobacterial cell wall: structure and role in natural resistance to antibiotics. *FEMS Microbiol Lett* **123**(1-2): 11-18.
- Kenneth, T. Tuberculosis, *Todar's Online Textbook of Bacteriology*. <http://textbookofbacteriology.net/tuberculosis.html>

- Kim, D., Kim, W. Y. ; Lee, S. Y. ; Lee, S. Y.; Yun, H.; Shin, S. Y.; Lee, J. ; Hong, Y. ; Won, Y. ; Kim, S. J.; Lee, Y. S. and Ahn, S. M. (2013). Revising a personal genome by comparing and combining data from two different sequencing platforms. *PLoS One* **8**(4): e60585.
- Kirschner, D. E.; Young, D. and Flynn, J. L. (2010). Tuberculosis: global approaches to a global disease. *Curr Opin Biotechnol* **21**(4): 524-531.
- Kitts, A. and Sherry, S. (2011). The Single Nucleotide Polymorphism Database (dbSNP) of Nucleotide Sequence Variation. The NCBI Handbook. <http://www.ncbi.nlm.nih.gov/books/NBK21088/>
- Kochi, A.; Varelzdis, B. and Styblo, K. (1993). Multidrug-resistant tuberculosis and its control. *Res Microbiol* **144**(2): 104-110.
- Korlach J. Pacific Biosciences:Overview. <http://www.pacificbiosciences.com>
- Kumar, P., Henikoff, S. and Ng, P. C. (2009). Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat Protoc* **4**(7): 1073-1081.
- Li, H. and Durbin, R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* **26**(5): 589-595.
- Li, H.; Ruan, J. and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Research* **18**(11): 1851-1858.
- Mardis, E. R. (2008). Next-generation DNA sequencing methods. *Annu Rev Genomics Hum Genet* **9**: 387-402.
- Margulies, M.; Egholm, M. ; Altman, W. E.; Attiya, S. ; Bader, J. S. ; Bembem, L. A. ; Berka, J. ; Braverman, M. S.; Chen, Y. J. ; Chen, Z. ; Dewell, S. B. ; Du, L. ; Fierro, J. M. ; Gomes, X. V. ; Godwin, B. C. ; He, W.; Helgesen, S. ; Ho, C. H. ; Irzyk, G. P. ; Jando, S. C.; Alenquer, M. L. ; Jarvie, T. P. ; Jirage, K. B. ; Kim, J. B. ; Knight, J. R. ; Lanza, J. R. ; Leamon, J. H. ; Lefkowitz, S. M.; Lei, M. ; Li, J. ; Lohman, K. L. ; Lu, H. ; Makhijani, V. B.; McDade, K. E.; McKenna, M. P.; Myers, E. W.; Nickerson, E.; Nobile, J. R.; Plant, R.; Puc, B. P.; Ronan, M. T.; Roth, G. T.; Sarkis, G. J.; Simons, J. F. ; Simpson, J. W.; Srinivasan, M.; Tartaro, K. R.; Tomasz, A.; Vogt, K. A.; Volkmer, G. A.; Wang, S. H.; Wang, Y.; Weiner, M. P.; Yu, P.; Begley, R. F. and Rothberg J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437**(7057): 376-380.
- McEvoy, C. R.; Cloete, R.; Muller, B.; Schurch, A. C.; Helden, P. D. van; Gagneux, S.; Warren, R. M. and Gey van Pittius, N. C. (2012). Comparative analysis of *Mycobacterium tuberculosis* *pe* and *ppe* genes reveals high sequence variation and an apparent absence of selective constraints. *PLoS One* **7**(4): e30593.
- Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nat Rev Genet* **11**(1): 31-46.
- Mitra, R. D.; Butty, V. L.; Shendure, J.; Williams, B. R.; Housman, D. E. and Church, G. M. (2003). Digital genotyping and haplotyping with polymerase colonies. *Proc Natl Acad Sci U S A* **100**(10): 5926-5931.

- Morris, R. P.; Nguyen, L.; Gatfield, J.; Visconti, K.; Nguyen, K.; Schnappinger, D.; Ehrh, S.; Liu, Y.; Heifets, L.; Pieters, J.; Schoolnik, G. and Thompson, C. J. (2005). Ancestral antibiotic resistance in *Mycobacterium tuberculosis*. *Proc Natl Acad Sci U S A* **102**(34): 12200-12205.
- Mostowy, S.; Cousins, D.; Brinkman, J.; Aranaz, A. and Behr, M. A. (2002). Genomic deletions suggest a phylogeny for the *Mycobacterium tuberculosis* complex. *J Infect Dis* **186**(1): 74-80.
- Musser, J. M. (1995). Antimicrobial agent resistance in mycobacteria: molecular genetic insights. *Clin Microbiol Rev* **8**(4): 496-514.
- Ng, P. C. and Henikoff, S. (2001). Predicting Deleterious Amino Acid Substitutions. *Genome Research* **11**(5): 863-874.
- Niemann, S.; Köser, C. U.; Gagneux, S.; Plinke, C.; Homolka, S.; Bignell, H.; Carter, R. J.; Cheetham, R. K.; Cox, A.; Gormley, N. A.; Kokko-Gonzales, P.; Murray, L. J.; Rigatti, R.; Smith, V. P.; Arends, F. P. M.; Cox, H. S.; Smith, G. and Archer, J. A. C. (2009). Genomic Diversity among Drug Sensitive and Multidrug Resistant Isolates of *Mycobacterium tuberculosis* with Identical DNA Fingerprints. *PLoS ONE* **4**(10): e7407.
- Ozsolak, F.; Kapranov, P.; Foissac, S.; Kim, S. W.; Fishilevich, E.; Monaghan, A. P.; John, B. and Milos, P. M. (2010). Comprehensive polyadenylation site maps in yeast and human reveal pervasive alternative polyadenylation. *Cell* **143**(6): 1018-1029.
- Pareek, C. S.; Smoczynski R. and Tretyn, A. (2011). Sequencing technologies and genome sequencing. *J Appl Genet* **52**(4): 413-435.
- Pellin, D., Miotto, P.; Ambrosi, A.; Cirillo, D. M. and Di Serio, C. (2012). A genome-wide identification analysis of small regulatory RNAs in *Mycobacterium tuberculosis* by RNA-Seq and conservation analysis. *PLoS One* **7**(3): e32723.
- Porreca, G. J.; Zhang, K.; Li, J. B.; Xie, B.; Austin, D.; Vassallo, S. L.; LeProust, E. M.; Peck, B. J.; Emig, C. J.; Dahl, F.; Gao, Y.; Church, G. M. and Shendure, J. (2007). Multiplex amplification of large sets of human exons. *Nat Methods* **4**(11): 931-936.
- Qi, W.; Käser, M.; Röltgen, K.; Yeboah-Manu, D. and Pluschke, G. (2009). Genomic Diversity and Evolution of *Mycobacterium ulcerans* Revealed by Next-Generation Sequencing. *PLoS Pathog* **5**(9): e1000580.
- Ramaswamy, S. and Musser, J. M. (1998). Molecular genetic basis of antimicrobial agent resistance in *Mycobacterium tuberculosis*: 1998 update. *Tuber Lung Dis* **79**(1): 3-29.
- Ramaswamy, S. V.; Reich, R.; Dou, S. J.; Jasperse, L.; Pan, X.; Wanger, A.; Quitugua, T. and Graviss, E. A. (2003). Single nucleotide polymorphisms in genes associated with isoniazid resistance in *Mycobacterium tuberculosis*. *Antimicrob Agents Chemother* **47**(4): 1241-1250.
- Reddy, T. B., Riley, R.; Wymore, F.; Montgomery, P.; DeCaprio, D.; Engels, R.; Gellesch, M.; Hubble, J.; Jen, D.; Jin, H.; Koehrsen, M.; Larson, L.; Mao, M.; Nitzberg, M.; Sisk, P.; Stolte, C.; Weiner, B.; White, J.; Zachariah, Z. K.; Sherlock, G.; Galagan, J. E.; Ball, C. A. and Schoolnik, G. K. (2009). TB database: an integrated platform for tuberculosis research. *Nucleic Acids Res* **37**(Database issue): D499-508.

- Riska, P. F.; Jacobs Jr., W. R. and Alland, D. (2000). Molecular determinants of drug resistance in tuberculosis. *Int J Tuberc Lung Dis* **4**(2 Suppl 1): S4-10.
- Rothberg, J. M.; Hinz, W.; Rearick, T. M.; Schultz, J.; Mileski, W.; Davey, M.; Leamon, J. H.; Johnson, K.; Milgrew, M. J.; Edwards, M.; Hoon, J.; Simons, J. F.; Marran, D.; Myers, J. W.; Davidson, J. F.; Branting, A.; Nobile, J. R.; Puc, B. P.; Light, D.; Clark, T. A.; Huber, M.; Branciforte, J. T.; Stoner, I. B.; Cawley, S. E.; Lyons, M.; Fu, Y.; Homer, N.; Sedova, M.; Miao, X.; Reed, B.; Sabina, J.; Feierstein, E.; Schorn, M.; Alanjary, M.; Dimalanta, E.; Dressman, D.; Kasinskas, R.; Sokolsky, T.; Fidanza, J. A.; Namsaraev, E.; McKernan, K. J.; Williams, A.; Roth, G. T. and Bustillo, J. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature* **475**(7356): 348-352.
- Rusk, N. (2009). Focus on next-generation sequencing data analysis. Forward. *Nat Methods* **6**(11 Suppl): S1.
- Sandgren, A.; Strong, M.; Muthukrishnan, P.; Weiner, B. K.; Church, G. M. and Murray, M. B. (2009). Tuberculosis drug resistance mutation database. *PLoS Med* **6**(2): e2.
- Sandgren, A.; Strong, M.; Muthukrishnan, P.; Weiner, B. K.; Church, G. M. and Murray, M. B. (2009). Tuberculosis Drug Resistance Mutation Database. *PLoS Medicine* **6**(2): e2.
- Sasseti, C. M. and Rubin, E. J. (2010). Relics of selection in the mycobacterial genome. *Nat Genet* **42**(6): 476-478.
- Sharma, D. and Surolia, A. (2011). Computational tools to study and understand the intricate biology of mycobacteria. *Tuberculosis (Edinb)* **91**(3): 273-276.
- Shendure, J., Porreca, G. J.; Reppas, N. B.; Lin, X.; McCutcheon, J. P.; Rosenbaum, A. M.; Wang, M. D.; Zhang, K.; Mitra, R. D. and Church, G. M. (2005). Accurate multiplex polony sequencing of an evolved bacterial genome. *Science* **309**(5741): 1728-1732.
- Silva, M. S., Senna, S. G.; Ribeiro, M. O.; Valim, A. R.; Telles, M. A.; Kritski, A.; Morlock, G. P.; Cooksey, R. C.; Zaha, A. and Rossetti, M. L. (2003). Mutations in *katG*, *inhA*, and *ahpC* genes of Brazilian isoniazid-resistant isolates of *Mycobacterium tuberculosis*. *J Clin Microbiol* **41**(9): 4471-4474.
- Sourceforge. <http://maq.sourceforge.net/index.shtml>
- Stein, L. D. (2010). The case for cloud computing in genome informatics. *Genome Biol* **11**(5): 207.
- Stucki, D. and Gagneux, S. (2013). Single nucleotide polymorphisms in *Mycobacterium tuberculosis* and the need for a curated database. *Tuberculosis (Edinb)* **93**(1): 30-39.
- Stoppler, M.C. (2011). Tuberculosis Symptoms, Causes, Treatment –Is there a vaccine against tuberculosis? http://www.medicinenet.com/tuberculosis/page_5.htm#tocg
- Takiff, H. E.; Salazar, L.; Guerrero, C.; Philipp, W.; Huang, W. M.; Kreiswirth, B.; Cole, S. T.; Jacobs Jr., W. R.; and Telenti, A. (1994). Cloning and nucleotide sequence of *Mycobacterium tuberculosis gyrA* and *gyrB* genes and detection of quinolone resistance mutations. *Antimicrob Agents Chemother* **38**(4): 773-780.
- Telenti, A. (1997). Genetics of drug resistance in tuberculosis. *Clin Chest Med* **18**(1): 55-64.

- Telenti, A.; Imboden, P.; Marchesi, F.; Schmidheini, T. and Bodmer, T. (1993). Direct, automated detection of rifampin-resistant *Mycobacterium tuberculosis* by polymerase chain reaction and single-strand conformation polymorphism analysis. *Antimicrob Agents Chemother* **37**(10): 2054-2058.
- The International SNP Map Working Group. (2001). A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*. **409**:928-933.
- Torres, T. T.; Metta, M.; Ottenwalder, B. and Schlotterer, C. (2008). Gene expression profiling by massively parallel sequencing. *Genome Res* **18**(1): 172-177.
- Wang, K.; Li, M. and Hakonarson, H. (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Research* **38**(16): e164-e164.
- Wetterstrand, K.A. (2013). DNA sequencing costs: data from the NHGRI large-scale genome sequencing program. National Human Genome Research Institute. <http://www.genome.gov/sequencingcosts/>
- Whiteford, N.; Haslam, N.; Weber, G.; Prugel-Bennett, A.; Essex, J. W.; Roach, P. L.; Bradley, M. and Neylon, C. (2005). An analysis of the feasibility of short read sequencing. *Nucleic Acids Res* **33**(19): e171.
- WHO.(2011). Global tuberculosis control. WHO Report. http://whqlibdoc.who.int/publications/2011/9789241564380_eng.pdf

9. APPENDIX

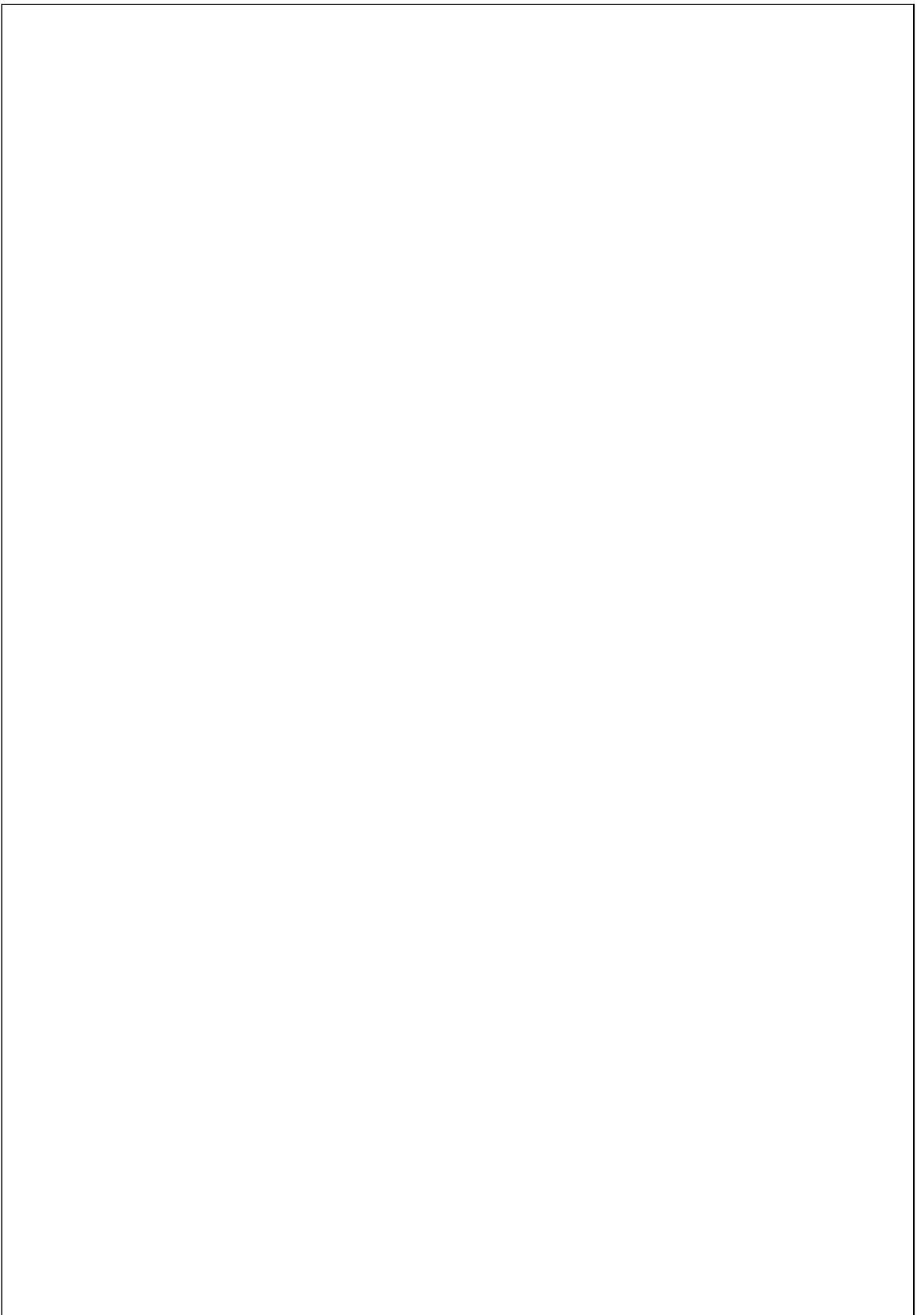
APPENDIX-I

Perl script to map Gene related information onto the Variants:

```
#!/usr/bin/perl

open(info1,"<map_exonic.txt") or die "cant open";
open(info2,"<gene.txt") or die "cant open";
open(info3,">map_exonic_gene.txt");

@a=<info1>;
@b=<info2>;
foreach $line1(@a)
{
    chomp $line1;
    @l1=split(/\t/,$line1);
    $gene="";
    $gene_name="";
    $RvID="";
    $start="";
    $stop="";
    $orientation="";
    $k=0;
    print"$l1[3]\n";
    foreach $line2(@b)
    {
        chomp $line2;
        @l2=split(/\t/,$line2);
        chomp $l1[3];chomp $l2[3]; chomp $l2[4];
        if($l1[3]>$l2[4] && $l1[3]<$l2[5])
        {
            print "$l1[3]\t$l2[3]\t$l2[4]\n";
            $gene_name=$l1[7];
            $RvID=$l2[1];
            $start,$l2[4];
            $stop,$l2[5];
            $orientation,$l2[3];
            $k++;
        }
        if(!$gene_name[1])
        {print info3 "$line1\t\t\t\t\t\n";}
        else
        {
            print info3 "$line1\t$gene_name\t$RvID\t$start\t$stop\t$orientation\n";
            print "$line1\t$gene_name\t$RvID\t$start\t$stop\t$orientation\n"; # into the output file along
            # with the prior information
        }
    }
}
```



APPENDIX-II

Perl script to map information related to regulatory regions onto the variants:

```
#!/usr/bin/perl

open(info1,"<map_all_var_gene.txt");
open(info2,"<peaks.txt");
open(info3,">map_var_gene_peaks.txt");

@a=<info1>;
@b=<info2>;
foreach $line1(@a)                                     ###Read the input file
{
    chop $line1;
    @l1=split(/\t/,$line1);
    @reg=""; @tar=""; @start=""; @stop=""; @type="";
    $k=0;
    foreach $line2(@b)                                   ###Read the file containing information
    regarding peaks
    {
        chomp $line2;
        @l2=split(/\t/,$line2);
        if($l1[3]>$l2[5] && $l1[3]<$l2[6])                 ###Checks whether the variant location falls within the
peak loci
        {
            ###In case the condition turns true the regulator, target names
and its
            push(@reg,$l2[0]);                             #start, stop positions and orientation are pushed to respective arrays
            push(@tar,$l2[1]);                             #or else the array remains empty
            push(@start,$l2[5]);
            push(@stop,$l2[6]);
            push(@type,$l2[9]);
            $k++;
        }
    }
    $regulator=join(';',@reg);
    $target=join(';',@tar);
    $start_pos=join(';',@start);
    $stop_pos=join(';',@stop);
    $type_pos=join(';',@type);
    if(!$reg[1])
    {
        chop $line1;
        print info3 "$line1\t\t\t\t\t\n";}
    else{
        chop $line1;
        print info3 "$line1\t$regulator\t$target\t$start_pos\t$stop_pos\t$type_pos\n";
        print "$line1\t$regulator\t$target\t$start_pos\t$stop_pos\t$type_pos\n";
    }
}
```

APPENDIX-III

Perl script to map ncRNA related information onto the Variants:

```
#!/usr/bin/perl

open(info1,"<map_var_gene_peaks.txt");
open(info2,"<nc_rna.txt");
open(info3,">map_var_gene_peaks_ncrna.txt");

@a=<info1>;
@b=<info2>;
foreach $line1(@a)                                     ###Read the Input file
{
  chomp $line1;
  @l1=split(/\t/,$line1);
  @start=""; @stop=""; @name=""; @strand=""; @product="";
  $k=0;
  print"$l1[2]\n";
  foreach $line2(@b)                                   ###Read the file containing info regarding ncRNA
  {
    chomp $line2;
    @l2=split(/\t/,$line2);
    chomp $l1[2];chomp $l2[2]; chomp $l2[3];
    if($l1[2]>$l2[2] && $l1[2]<$l2[3])                 ###Check whether the variant position fall within
the ncRNA
    {
      #genomic loci or not
      print "$l1[2]\t$l2[2]\t$l2[3]\n";                #In case the condition turns true, the required info is fetched
      push(@start,$l2[2]);                             #and pushed into the corresponding arrays
      push(@stop,$l2[3]);                               #or else the arrays remain empty
      push(@name,$l2[4]);
      push(@strand,$l2[6]);
      push(@product,$l2[7]);
      $k++;
    }
  }
  if(!$start[1])
  {print info3 "$line1\t\t\t\t\t\n";}
  else
  {
    print info3 "$line1\t$start[1]\t$stop[1]\t$name[1]\t$strand[1]\t$product[1]\n";
    print "$line1\t$start[1]\t$stop[1]\t$name[1]\t$strand[1]\t$product[1]\n";
  }
}
```

APPENDIX-IV

CGI script to extract the input query fed by the user, retrieve the corresponding information from the compiled datasheet and represent it separately under different tabs:

```
#!/usr/bin/perl
##### Importing libraries#####
use strict;
use CGI qw(:standard);
use DBI;
use List::MoreUtils qw/ uniq /;

#####Fetching query fed by the user#####
my $cgi= new CGI;
my $query=$cgi->param('query');
my $table_nm = $cgi->param('hidden');
my $pos=""; my $rvid; my $name; my $range;

print "Content-type:text/html\r\n\r\n";

#####Coding style-sheet#####
print '<html><head>
<style>

#tb {color:#003300;}
#var {color:#FFFFFF;}

#Tables li:link { text-decoration:none; background-color:#003366;color:#FFFFFF;border-radius:4px }
#Tables li:hover { text-decoration:none; background-color:gray;color:#FFFFFF;border-radius:4px }
#Tables li:visited { text-decoration:none; color:#FFFFFF;}

table a:link {color:#4682B4; text-decoration:none; font-size: 13px; font-weight:bold;}
table a:visited {color:#4682B4; text-decoration:none; font-size: 13px; font-weight:bold;}
table a:hover {color:#191970; text-decoration:none; font-size: 13px;font-weight:bold;}

a.'$table_nm.':link, a.'$table_nm.':visited{color:##FFCC00; text-decoration:none; font-size:125%;
background-color:#FF9900; border-style:solid;border-width:0px 0px 3.5px 0px ;border-
color:#FFFFFF; border-radius: 4px; font-weight:bold;}
a.'$table_nm.':hover,{color:##FFCC00; text-decoration:none; font-size:125%; background-
color:yellow; border-style:solid;border-width:0px 0px 3.5px 0px ;border-color:#FFFFFF; border-
radius: 4px; font-weight:bold;}

#Links a:link {color:#003366; text-decoration:none;}
#Links a:hover{color:#FFCC00; text-decoration:none;}

table tr:hover {color:#000000; background-color:#E6E6FA;}

body{background-color:#FFFFFF; margin-left: 10px; margin-right: 10px;}
#Header{color:#191970; padding: 0px 0px 0px 30px; background-color: #003366;}
hr{color:#FFFFFF;}

#headertab td, th { border:1px solid #003366; }
#header table {border-collapse: collapse; border:1px solid #003366;}
#headertab th {background-color: #003366;}
#headertab td {background-color: #003366;}

#Links{background-color:#FFFFFF}
#Search{background-color:#FFFFFF;font-size:15pt;}
```

```
#Search input{border-style:solid;border-width:1.5px;border-color:#003366;border-radius: 4px;}
```

```
#Tables{font-size:15pt;border-radius: 4px;max-width:95%;}  
#Tables ul {list-style: none;margin:0;padding:0;margin-bottom:0;}  
#Tables li {display: inline;margin: 0 0.5em 0 0;}
```

```
#Result{color:#4682B4;background-color:#FFFFFF;font-weight:bold;max-width:75%;margin-left:15%;border-style:solid;border-width:1.5px 1.5px 1.5px 1.5px;border-color:gray;border-radius: 4px;}
```

```
table {color:#44677D;border-collapse:collapse; min-width:90%;font-size:14px;}  
td {font-size:14px;border-collapse:collapse;text-align:center;padding:6px 6px 6px 6px;}  
th{background-color:#003366;color:#FFFFFF;border-color:#FFFFFF;font-size:16px;}  
table tr:nth-child(odd) td{background-color:#FFFFFF;}  
table tr:nth-child(even) td{background-color:#E6E6FA;}
```

```
#footer{color:#FFFFFF;font-size:10pt; padding: 10px 10px; background-color:#003366;border-radius: 4px;}  
</style>
```

```
<title>tbvar</title>  
</head>
```

```
##### Designing Result page of the interface#####
```

```
<body style="font-family: Arial;">  
<div id=Header style="vertical-align:middle;height:100;">  
<a href="/Index.html" title="Go to tbvar Home Page" ></a></div>  
<br>  
<div id=Links><b>  
<a href="/Index.html" ></a>  
<a href="/annoTB.html" ></a>  
<a href="/help.html" ></a>  
<a href="/contact.html"></a>  
</b>  
</div>
```

```
<div id=Search>  
<br><center><form id="submit" action="tbvar.cgi" method="Get" align="center"><input type="text" id="query" name="query" size=35% placeholder="Search: Variant Location, Range, Gene or Rvid" ><input name="hidden" type="hidden" value="variation">&nbsp;&nbsp; <input type="submit" value="Eureka"></center></form></div>;
```

```
#####Connecting to the database#####  
my $dbh=DBI->connect('dbi:mysql:mtb_anno','root','123456') or die "Connection Error:$DBI::errstr\n";
```

```
#####Checking the type of query fed#####
```

```
if($query=~/\d*\-\d*/)  
{  
$range=$query;  
}  
  
if($query=~^[0-9]*$/)  
{  
$pos=$query;  
}  
  
if($query=~^[Rr]v*/)  
{  
$rvid=$query;}
```

```

if($query=~/[a-z A-Z]+)/
{
$name=$query;}

#####----Tabs-----#####
print "      <center>
<div id=Tables><ul id='nav'><br>
<li id='tab1'><a class='variation' href='/cgi-bin/tbvar.cgi?query=$query&hidden=variation' class='links'
id='link1' title='Shows the basic information of the variations including type frequency and mapping to other
resources'><img src=/genomic_variation.gif></a></li>
<li id='tab4'><a class = 'Gene' href='/cgi-bin/tbvar.cgi?query=$query&hidden=Gene' id='link4' class='links'
title='Shows information pertaining to the gene housing the variations'><img
src=/gene_annotation.gif></a></li>
<li id='tab3'><a class = 'non_syn_mutant_gene' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=non_syn_mutant_gene' class='links' id='link3' title='Shows information on
functional consequence of a variation on the protein'><img src=/Functional_effects.gif></a></li>
<li id='tab5'><a class = 'Regulatory_Variations' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=Regulatory_Variations' class='links' id='link5' title='Shows information on
mapping of the variations on the Transcription factor regulatory elements'><img
src=/regulatory_variations.gif></a></li>
<li id='tab7'><a class = 'variant_strain_info' href='/cgi-bin/tbvar.cgi?query=$query&hidden=variant_strain_info'
class='links' id='link7' title='Shows information on the sequencing of the sample, experiment and the study from
which the variation was derived'><img src=/strain_information.gif></a></li>;

my $sql1="Select * from mtb_anno.tbvar_db";
my $sql2="Select * from mtb_anno.drug";
if($rvid)
{
    my $sql1_ap=$sql1." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
    my $sth1=$dbh->prepare($sql1_ap);
    $sth1->execute or die "SQL error:$DBI::errstr\n";
    my @all_1= $sth1->fetchrow_array;
    my $rvid_drug= $all_1[8];

    my $sql2_ap=$sql2." where UPPER(var_resis_gene)=UPPER(\"$rvid_drug\")";
    my $sth2=$dbh->prepare($sql2_ap);
    $sth2->execute or die "SQL error:$DBI::errstr\n";
    my @all_2= $sth2->fetchrow_array;

if($all_1[24])
{print"<li id='tab8'><a class = 'nc_rna' href='/cgi-bin/tbvar.cgi?query=$query&hidden=nc_rna' class='links'
id='link8' title='Shows information on variations mapping to ncRNA loci of the genome'><img
src=/ncRNA.gif></a></li>"}
if($all_2[1])
{print"<li id='tab6'><a class = 'Drug_Resistance' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=Drug_Resistance' class='links' id='link6' title='Shows information on
variations found to confer drug resistance property to Mtb'><img src=/drugresist.gif></a></li>"}
}
if($range)
{
    my @ran=split(/\-/, $range);
    my $start=$ran[0];
    my $stop=$ran[1];
    my $sql1_ap=$sql1." where var_int_id between \"$start\" and \"$stop\"";
    my $sql2_ap=$sql2." where var_int_id between \"$start\" and \"$stop\"";
    my $sth1=$dbh->prepare($sql1_ap);
    my $sth2=$dbh->prepare($sql2_ap);
    $sth1->execute or die "SQL error:$DBI::errstr\n";
    $sth2->execute or die "SQL error:$DBI::errstr\n";
    my @all_1= $sth1->fetchrow_array;

```

```

        my @all_2= $sth2->fetchrow_array;
if($all_1[24])
{print"<li id='tab8'><a class = 'nc_rna' href='/cgi-bin/tbvar.cgi?query=$query&hidden=nc_rna' class='links'
id='link8' title='Shows information on variations mapping to ncRNA loci of the genome'><img
src=/ncRNA.gif></a></li>" }
if($all_2[1])
{print"<li id='tab6'><a class = 'Drug_Resistance' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=Drug_Resistance' class='links' id='link6' title='Shows information on
variations found to confer drug resistance property to Mtb'><img src=/drugresist.gif></a></li>" }
}
if($pos)
{
    my $sql1_ap=$sql1." where var_int_id=\"$pos\"";
    my $sql2_ap=$sql2." where var_int_id=\"$pos\"";
    my $sth1=$dbh->prepare($sql1_ap);
    my $sth2=$dbh->prepare($sql2_ap);
    $sth1->execute or die "SQL error:$DBI::errstr\n";
    $sth2->execute or die "SQL error:$DBI::errstr\n";
    my @all_1= $sth1->fetchrow_array;
    my @all_2= $sth2->fetchrow_array;
if($all_1[24])
{print"<li id='tab8'><a class = 'nc_rna' href='/cgi-bin/tbvar.cgi?query=$query&hidden=nc_rna' class='links'
id='link8' title='Shows information on variations mapping to ncRNA loci of the genome'><img
src=/ncRNA.gif></a></li>" }
if($all_2[1])
{print"<li id='tab6'><a class = 'Drug_Resistance' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=Drug_Resistance' class='links' id='link6' title='Shows information on
variations found to confer drug resistance property to Mtb'><img src=/drugresist.gif></a></li>" }
}
if($name)
{
    my $sql1_ap=$sql1." where UPPER(gene_name)=UPPER(\"$name\")";
    my $sql2_ap=$sql2." where UPPER(var_resis_gene)=UPPER(\"$name\")";
    my $sth1=$dbh->prepare($sql1_ap);
    my $sth2=$dbh->prepare($sql2_ap);
    $sth1->execute or die "SQL error:$DBI::errstr\n";
    $sth2->execute or die "SQL error:$DBI::errstr\n";
    my @all_1= $sth1->fetchrow_array;
    my @all_2= $sth2->fetchrow_array;
if($all_1[24])
{print"<li id='tab8'><a class = 'nc_rna' href='/cgi-bin/tbvar.cgi?query=$query&hidden=nc_rna' class='links'
id='link8' title='Shows information on variations mapping to ncRNA loci of the genome'><img
src=/ncRNA.gif></a></li>" }
if($all_2[1])
{print"<li id='tab6'><a class = 'Drug_Resistance' href='/cgi-
bin/tbvar.cgi?query=$query&hidden=Drug_Resistance' class='links' id='link6' title='Shows information on
variations found to confer drug resistance property to Mtb'><img src=/drugresist.gif></a></li>" }
}
print"<li id='tab9'><a class = 'Browser' href='/cgi-bin/tbvar.cgi?query=$query&hidden=Browser' class='links'
id='link8' title='Browse the region surrounding the variation and genes harbouring the variations'><img
src=/genome_browser.gif></a></li>
</div></center>";

if($query)
{print"<div id=Result><center>";}

#####-----For Variation Table-----#####
if($table_nm eq "variation")
{

```

```

print"<center><br><table><tr><th>Gene Id&nbsp;&nbsp;&nbsp;</th><th>Position&nbsp;&nbsp;&nbsp;</th><th>Ref
Allele&nbsp;&nbsp;&nbsp;</th><th>Alt
Allele&nbsp;&nbsp;&nbsp;</th><th>Location&nbsp;&nbsp;&nbsp;</th><th>Type&nbsp;&nbsp;&nbsp;</th><th>Variant
Count&nbsp;&nbsp;&nbsp;</th><th>Frequency Percentage&nbsp;&nbsp;&nbsp;</th><th>External
Link&nbsp;&nbsp;&nbsp;</th></tr><tr>";
my $sql1="Select gene_int_id,var_int_id,var_ref,var_alt,var_loci,var_type,var_freq,var_snp_id from
mtb_anno.tbvar_db";
if($rvid)
{
    my $sql1_ap=$sql1." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
    &var($sql1_ap);
}
if($range)
{
    my @ran=split(/-/, $range);
    my $start=$ran[0];
    my $stop=$ran[1];
    my $sql1_ap=$sql1." where var_int_id between \"$start\" and \"$stop\"";
    &var($sql1_ap);
}
if($pos)
{
    my $sql1_ap=$sql1." where var_int_id=\"$pos\"";
    &var($sql1_ap);
}
if($name)
{
    my $sql1_ap=$sql1." where UPPER(gene_name)=UPPER(\"$name\")";
    &var($sql1_ap);
}
}

#####-----For Gene Table-----#####
if($table_nm eq "Gene")
{
    my $sql4="Select gene_int_id, gene_name, gene_anno, gene_start, gene_stop, gene_orientation from
mtb_anno.tbvar_db";
    print"<center><br><table><th>Gene Id </th><th>Gene Name</th><th>Gene
Annotation</th><th>Start</th><th>Stop</th><th>Orientation</th></tr><tr>";
    if($rvid eq $name)
    {
        my $sql4_ap=$sql4." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
        my $sth4=$dbh->prepare($sql4_ap);
        $sth4->execute or die "SQL error:$DBI::errstr\n";
        my @all=$sth4->fetchrow_array;

        if($all[0])
        {print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation'
id='link1'>$all[0]</a></td><td>
$all[1]</td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td></tr><tr></center>";}
    }
else
{
if($rvid)
{
    my $sql4_ap=$sql4." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
    my $sth4=$dbh->prepare($sql4_ap);
    $sth4->execute or die "SQL error:$DBI::errstr\n";
}
}
}

```



```

        my @ran=split(/\-/, $range);
        my $start=$ran[0];
        my $stop=$ran[1];
        my $sql3_ap=$sql3." where var_int_id between \" $start\" and \" $stop\"";
        &nsmg($sql3_ap);
    }
    if($name)
    {
        my $sql3_ap=$sql3." where UPPER(gene_name)=UPPER(\" $name\")";
        &nsmg($sql3_ap);
    }
}

#####-----For Drug Resistance Table-----#####
if($table_nm eq "Drug_Resistance")
{
    print"<br><table><tr><th>Position&nbsp;&nbsp;</th><th>Resistant Drug</th><th>Resistant
Gene</th><th>Reference</th></tr><tr><tr>";
    my $sql5="Select * from mtb_anno.drug";
    if($rvid)
    {
        my $sql="Select * from mtb_anno.tbvar_db where gene_int_id=\" $rvid\"";
        my $sth=$dbh->prepare($sql);
        $sth->execute or die "SQL error:$DBI::errstr\n";
        my @all=$sth->fetchrow_array;
        my $sql5_ap=$sql5." where UPPER(var_resis_gene)=UPPER(\" $all[8]\");";
        &drug($sql5_ap);
    }
    if($pos)
    {
        my $sql5_ap=$sql5." where var_int_id=\" $pos\"";
        &drug($sql5_ap);
    }
    if($range)
    {
        my @ran=split(/\-/, $range);
        my $start=$ran[0];
        my $stop=$ran[1];
        my $sql5_ap=$sql5." where var_int_id between \" $start\" and \" $stop\"";
        &drug($sql5_ap);
    }
    if($name)
    {
        my $sql5_ap=$sql5." where UPPER(var_resis_gene)=UPPER(\" $name\")";
        &drug($sql5_ap);
    }
}

#####-----For Regulatory Variations Table-----#####
if($table_nm eq "Regulatory_Variations")
{
    print"<br><table><tr><th>Position&nbsp;&nbsp;</th><th>Regulator</th><th>Target</th><th>Start</th><th>
Stop</th></tr><tr><tr>";
    my $sql6="Select * from mtb_anno.tbvar_db";
    if($rvid)
    {
        my $sql6_ap=$sql6." where UPPER(gene_int_id)=UPPER(\" $rvid\")";
        &reg($sql6_ap);
    }
    if($pos)

```

```

{
    my $sql6_ap=$sql6." where var_int_id=\"$pos\"";
    &reg($sql6_ap);
}
if($range)
{
    my @ran=split(/\-/, $range);
    my $start=$ran[0];
    my $stop=$ran[1];
    my $sql6_ap=$sql6." where var_int_id between \"$start\" and \"$stop\"";
    &reg($sql6_ap);
}
if($name)
{
    my $sql6_ap=$sql6." where UPPER(gene_name)=UPPER(\"$name\")";
    &reg($sql6_ap);
}
}

```

#####-----For Variant Strain Info Table-----#####

```

if($table_nm eq "variant_strain_info")
{
    print"<br><table><tr><th>Position&nbsp;&nbsp;&nbsp;</th><th>Sample</th><th>Experiment</th><th>
Reference</th></tr><tr><tr>";
    my $sql7="Select * from mtb_anno.tbvar_db";
    if($rvid)
    {
        my $sql7_ap=$sql7." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
        &strain($sql7_ap);
    }
    if($pos)
    {
        my $sql7_ap=$sql7." where var_int_id=\"$pos\"";
        &strain($sql7_ap);
    }
    if($range)
    {
        my @ran=split(/\-/, $range);
        my $start=$ran[0];
        my $stop=$ran[1];
        my $sql7_ap=$sql7." where var_int_id between \"$start\" and \"$stop\"";
        &strain($sql7_ap);
    }
    if($name)
    {
        my $sql7_ap=$sql7." where UPPER(gene_name)=UPPER(\"$name\")";
        &strain($sql7_ap);
    }
}
}

```

#####-----For ncRNA-----#####

```

if($table_nm eq "nc_rna")
{
    print"<br><table><tr><th>Position&nbsp;&nbsp;&nbsp;</th><th>ncRNA Start</th><th>ncRNA
Stop</th><th>ncRNA Name</th><th>ncRNA strand</th><th>ncRNA Product</th></tr><tr>";
    my $sql8="Select * from mtb_anno.tbvar_db";
    if($rvid)
    {
        my $sql8_ap=$sql8." where UPPER(gene_int_id)=UPPER(\"$rvid\")";
        &ncrna($sql8_ap);
    }
}

```

```

}
if($pos)
{
    my $sql8_ap=$sql8." where var_int_id=\" $pos\"";
    &ncrna($sql8_ap);
}
if($range)
{
    my @ran=split(/\-/, $range);
    my $start=$ran[0];
    my $stop=$ran[1];
    my $sql8_ap=$sql8." where var_int_id between \" $start\" and \" $stop\"";
    &ncrna($sql8_ap);
}
if($name)
{
    my $sql8_ap=$sql8." where UPPER(gene_name)=UPPER(\" $name\")";
    &ncrna($sql8_ap);
}
}

```

#####-----For Browser-----#####

```

if($table_nm eq "Browser")
{
    if($name || $rvid)
    {
        if($name eq $rvid)
        {
            my $sql="Select * from mtb_anno.tbvar_db where UPPER(gene_name)=UPPER(\" $name\")";
            my $sth=$dbh->prepare($sql);
            $sth->execute or die "SQL error:$DBI::errstr\n";
            my @all;
            @all=$sth->fetchrow_array;
            my $gene_start=$all[9];
            my $gene_stop=$all[10];
            print"<br><iframe style='border: 1px solid black; margin-left:25px; margin-right:50px;'
            src='/jbrowse/index.html?data=mtb/json&loc=chr1%3A$all[9]..$all[10]&tracks=DNA%2CGenes%2CVariation
            s' width='900' height='400' ></iframe>";
        }
        else
        {
            if($name)
            {
                my $sql="Select * from mtb_anno.tbvar_db where UPPER(gene_name)=UPPER(\" $name\")";
                my $sth=$dbh->prepare($sql);
                $sth->execute or die "SQL error:$DBI::errstr\n";
                my @all;
                @all=$sth->fetchrow_array;
                my $gene_start=$all[9];
                my $gene_stop=$all[10];
                print"<br><iframe style='border: 1px solid black; margin-left:25px; margin-right:50px;'
                src='/jbrowse/index.html?data=mtb/json&loc=chr1%3A$all[9]..$all[10]&tracks=DNA%2CGenes%2CVariation
                s' width='900' height='400' ></iframe>";
            }
        }
        if($rvid)
        {
            my $sql="Select * from mtb_anno.tbvar_db where UPPER(gene_int_id)=UPPER(\" $rvid\")";
            my $sth=$dbh->prepare($sql);
            $sth->execute or die "SQL error:$DBI::errstr\n";
            my @all;

```

```

@all=$sth->fetchrow_array;
my $gene_start=$all[9];
my $gene_stop=$all[10];
print "$all[8]\t$all[7]\t$gene_start\t$gene_stop<br>";
print"<br><iframe style='border: 1px solid black; margin-left:25px; margin-right:50px;'
src='/jbrowse/index.html?data=mtb/json&loc=chrI%3A$all[9]..$all[10]&tracks=DNA%2CGenes%2CVariation
s' width='900' height='400' ></iframe>";
}
}
}
if($pos)
{
print"<br><iframe style='border: 1px solid black; margin-left:25px; margin-right:50px;'
src='/jbrowse/index.html?data=mtb/json&loc=chrI%3A$pos&tracks=DNA%2CGenes%2CVariations'
width='900' height='400' ></iframe>";
}
if($range)
{
my @ran=split(/\-/, $range);
my $start=$ran[0];
my $stop=$ran[1];
print"<br><iframe style='border: 1px solid black; margin-left:25px; margin-right:50px;'
src='/jbrowse/index.html?data=mtb/json&loc=chrI%3A$start..$stop&tracks=DNA%2CGenes%2CVariations'
width='900' height='400' ></iframe>";
}
}
print "</tr></table><br></div><br></center>";

if(!$query)
{print"<br><br><br><br><br><br><br><br><br><br><br><br><br><br>";}
print"</center></body></html>";

```

#####-----Subroutines to show result-----#####

```

sub var
{
my($sql)=@_;
my $sth1=$dbh->prepare($sql);
$sth1->execute or die "SQL error:$DBI::errstr\n";
my @all;

while (@all = $sth1->fetchrow_array)
{
my $sql_drug="Select * from mtb_anno.drug where var_int_id=$all[1]";
my $sth_drug=$dbh->prepare($sql_drug);
$sth_drug->execute or die "SQL error:$DBI::errstr\n";
my @all_drug = $sth_drug->fetchrow_array;

#print"$all[1]<br>";
my $freq_pct=sprintf "%.2f",((($all[6]/469)*100);
chop $all[7];
if($all[7] =~ /^rs[0-9]+$/)
{print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene'
id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation'
id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td>
<td>$freq_pct</td><td><a
href='http://www.ncbi.nlm.nih.gov/projects/SNP/snp_ref.cgi?searchType=ad hoc_search&type=rs&rs=$all[7]'>d
bSNP</a></td></tr><tr></center>";next;}
if($all[7] =~ /^rs[0-9]+ ;7000[0-9]+/)
{my @links=split(/;/,$all[7]);

```

```

print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td><a href='http://www.ncbi.nlm.nih.gov/projects/SNP/snp_ref.cgi?searchType=ad hoc_search&type=rs&rs=$links[0]>dbSNP</a>;<a href='http://genome.tdbb.org/annotation/genome/tbdb/FeatureDetails.html?sp=$links[1]&sp=SPolyLocus'>TBDB</a></td></tr><tr></center>";next;}
if($all[7]=~/^7000[0-9]+$/)
{print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td><a href='http://genome.tdbb.org/annotation/genome/tbdb/FeatureDetails.html?sp=$all[7]&sp=SPolyLocus'>TBDB</a></td></tr><tr></center>";next;}
if($all[7]=~/^mtci/)
{print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td><a href='http://ccb.jnu.ac.in/cgi-bin/mtcid/search'>MTCID</a></td></tr><tr></center>";next;}
if($all[7]=~/^rs[0-9]+ ;mtci$/)
{my @links=split(/;/,$all[7]);
print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td><a href='http://www.ncbi.nlm.nih.gov/projects/SNP/snp_ref.cgi?searchType=ad hoc_search&type=rs&rs=$links[0]>dbSNP</a>;<a href='http://ccb.jnu.ac.in/cgi-bin/mtcid/search'>MTCID</a></td></tr><tr></center>";next;}
if($all[7]=~/7000[0-9]+;mtci$/)
{my @links=split(/;/,$all[7]);
print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td><a href='http://genome.tdbb.org/annotation/genome/tbdb/FeatureDetails.html?sp=$all[7]&sp=SPolyLocus'>TBDB</a>;<a href='http://ccb.jnu.ac.in/cgi-bin/mtcid/search'>MTCID</a></td></tr><tr></center>";next;}
else
{print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=Gene' id='link4'>$all[0]</a></td><td><a href='/cgi-bin/tbvar.cgi?query=$all[1]&hidden=variation' id='link1'>$all[1]</a></td><td>$all[2]</td><td>$all[3]</td><td>$all[4]</td><td>$all[5]</td><td>$all[6]</td><td>$freq_pct</td><td></td></tr><tr></center>";}
}
}
sub gene
{
my($sql)=@_;
my $sth4=$dbh->prepare($sql);
$sth4->execute or die "SQL error:$DBI::errstr\n";
my @all; my @name="";
while(@all=$sth4->fetchrow_array)
{
push(@name,$all[0]);
}
my @unique = uniq @name;
my $l=@unique;
for(my $i=1;$i<$l;$i++)
{
my $sql_un="Select gene_int_id, gene_name, gene_anno, gene_start, gene_stop, gene_orientation
from mtb_anno.tbvar_db where UPPER(gene_name)=UPPER(\"$unique[$i]\")";

```

```

        my $sth4_un=$dbh->prepare($sql_un);
        $sth4_un->execute or die "SQL error:$DBI::errstr\n";
        my @all_un=$sth4_un->fetchrow_array;
        print"<center><td><a href='/cgi-bin/tbvar.cgi?query=$all_un[1]&hidden=variation' id='link1'>
$all_un[0]</a></td><td>$all_un[1]</td><td>$all_un[2]</td><td>$all_un[3]</td><td>$all_un[4]</td><td>$all_
un[5]</td></tr></center>";
    }
}

```

sub nsmg

```

{
my($sql)=@_;
my $sth3=$dbh->prepare($sql);
$sth3->execute or die "SQL error:$DBI::errstr\n";

my @all;my @mut;
while(@all=$sth3->fetchrow_array)
{
push(@mut,$all[12]);my $l=@mut; $l=$l-1;my $sl=$l-1;
if($l>0 && $mut[$l] eq $mut[$sl])
{next;}
else
{
if($all[12])
{
if(!$all[13])
{ $all[13]='Not Scored'; $all[14]='Not Scored';
print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$all[12]</td><td>$all[13]</td><td>$all[14]</td></tr><tr>";}
else
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$all[12]</td><td>$all[13]</td><td>$all[14]</td></tr><tr>";}
}
else
{ $all[12]='No Mutation';
if(!$all[13])
{ $all[13]='Not Scored'; $all[14]='Not Scored';
print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$all[12]</td><td>$all[13]</td><td>$all[14]</td></tr><tr>";}
else
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$all[12]</td><td>$all[13]</td><td>$all[14]</td></tr><tr>";}
}
}
}
}
}

```

sub drug

```

{
my($sql)=@_;
my $sth5=$dbh->prepare($sql);
$sth5->execute or die "SQL error:$DBI::errstr\n";
my @all;
while(@all=$sth5->fetchrow_array)
{
print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$all[1]</td><td>$all[2]</td><td>$all[3]</td></tr><tr>";
}
}

```

```

sub reg
{
my($sql)=@_;
my $sth6=$dbh->prepare($sql);
$sth6->execute or die "SQL error:$DBI::errstr\n";
my @all; my @pos;my $l_pos;

while(@all=$sth6->fetchrow_array)
{
push(@pos,$all[0]);my $l=@pos; $l=$l-1;my $sl=$l-1;
if($l>0 && $pos[$l]==$pos[$sl])
{next;}
else
{#print"i m here $pos[$l_pos]<br>$pos[$l_pos]:$pos[$l_pos-1]";
if(!$all[15])
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>No Regulatory Effect</td><td></td><td></td><td></td></tr><tr>";}
else
{
my @reg=split(/;,$all[15]);
my @tar=split(/;,$all[16]);
my @start=split(/;,$all[17]);
my @stop=split(/;,$all[18]);
my $len=@reg;
for(my $i=1;$i<$len;$i++)
{
if($rvid eq $tar[$i])
{
print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
$all[0]</a></td><td>$reg[$i]</td><td>$tar[$i]</td><td>$start[$i]</td><td>$stop[$i]</td></tr><tr>";
}
if($range)
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
@all[0]</a></td><td>$reg[$i]</td><td>$tar[$i]</td><td>$start[$i]</td><td>$stop[$i]</td></tr><tr>";}
if($pos)
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
@all[0]</a></td><td>$reg[$i]</td><td>$tar[$i]</td><td>$start[$i]</td><td>$stop[$i]</td></tr><tr>";}
if($name)
{
if($all[7] eq $tar[$i])
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
@all[0]</a></td><td>$reg[$i]</td><td>$tar[$i]</td><td>$start[$i]</td><td>$stop[$i]</td></tr><tr>";}
}
}
}
}next;
}}
sub strain
{
my($sql)=@_;
my $sth7=$dbh->prepare($sql);
$sth7->execute or die "SQL error:$DBI::errstr\n";
my @all;my @pos;
while(@all=$sth7->fetchrow_array)
{
push(@pos,$all[0]);my $l=@pos; $l=$l-1;my $sl=$l-1;

my @sample=split(/;,$all[19]);
my @exp=split(/;,$all[20]);
my @ref=split(/;,$all[21]);

```

```

my $len=@sample;
for(my $i=0;$i<$len;$i++)
{print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'> @all[0]</a></td><td><a
href='http://sra.dnanexus.com/samples/$sample[$i]'>$sample[$i]</a></td><td><a
href='http://sra.dnanexus.com/experiments/$exp[$i]'>$exp[$i]</a></td><td><a
href='http://sra.dnanexus.com/studies/$ref[$i]'>$ref[$i]</a></td></tr><tr>";}

}
}
sub ncrna
{
my($sql)=@_;
my $sth7=$dbh->prepare($sql);
    $sth7->execute or die "SQL error:$DBI::errstr\n";
    my @all;my @pos;
    while(@all=$sth7->fetchrow_array)
    {
    push(@pos,$all[0]);my $l=@pos; $l=$l-1;my $sl=$l-1;
    if($l>0 && $pos[$l]==$pos[$sl])
    {next;}
    else
    {
    if($all[24])
    {print"<td><a href='/cgi-bin/tbvar.cgi?query=$all[0]&hidden=variation' id='link1'>
    $all[0]</a></td><td>$all[23]</td><td>$all[24]</td><td>$all[25]</td><td>$all[26]</td><td>$all[27]</td></tr>
    <tr>";}
    }
    }
}
}
print"</tr></table><br><center></center></div><br><br><br><br><br><br>
<div id=footer style='height:30px;margin-right:5px;width=100; color:#FFFFFF;font-size:10pt; padding: 10px
10px; background-color:#003366;'><b>Copyright&copy;2013 CSIR</b></div>";
#####

```

APPENDIX-V

CGI script to extract the inserted variant file in the annoTB web-page, retrieve the corresponding information from the compiled datasheet and present in the form of a report

```
#!/usr/bin/perl

use strict;
use CGI;
use DBI;
use List::MoreUtils qw/ uniq /;

my $upload_dir = "/usr/lib/cgi-bin/upload";

#####Extractingthe query submitted in the form#####
my $cgi = new CGI;
my $query_batch=$cgi->param('query');
my $table_nm = $cgi->param('hidden');
my $t = $cgi->param("batch");
my @text=split(/\n/,$t);

my @input;my $query;my @ref;my @alt;my $in_len;

if(@text)
{
  @input="";@ref="";@alt="";
  foreach my $textline(@text)
  {
    my @val=split(/\s+/, $textline);
    $val[1]=~s/\s//g;
    push(@input,$val[0]);
    push(@ref,$val[1]);
    push(@alt,$val[2]);
  }
  $in_len=@input;
  $query=join(" ",@input);
  $query=substr($query,1);
}

#####Designing the web page#####
print "Content-type:text/html\r\n\r\n";
print '<html><head>
#####Coding the style-sheet#####
<style>

#tb {color:#003300;}
#var {color:#FFFFFF;}

#Result_Table tr:hover {color:#000000; background-color:#E6E6FA;}
#Result_Novel tr:hover {color:#000000; background-color:#E6E6FA;}

body{background-color:#FFFFFF; margin-left: 10px; margin-right: 10px;}
#Header{color:#191970;font-size:30pt; padding: 0px 0px 0px 30px; background-color: #003366;}
hr{color:#FFFFFF;}

p{color:#003366;}
```

```

#Links{background-color:#FFFFFF}
#Search{background-color:#FFFFFF;font-size:15pt;}
#Search input{border-style:solid;border-width:1.5px;border-color:#003366;border-radius: 4px;}

#Result{color:#4682B4;background-color:#FFFFFF;font-weight:bold;max-width:55%;margin-
left:18%;border-style:solid;border-width:1.5px 1.5px 1.5px 1.5px;border-color:#003366;border-radius:
4px;padding:0px 30px 0px 85px;}

table {color:#44677D;border-collapse:collapse; min-width:80%;font-size:17px;text-align:center;}
table a:link {color:#4682B4; text-decoration:none; font-size: 12px; font-weight:bold;}
table a:visited {color:#4682B4; text-decoration:none; font-size: 12px; font-weight:bold;}
table a:hover {color:#191970; text-decoration:none; font-size: 12px; font-weight:bold;}

#Result_Table td {font-size:14px;border-collapse:collapse;text-align:center;}
#Result_Table th{background-color:#003366;color:#FFFFFF;border-color:#FFFFFF;text-
align:center;}
#Result_Table table tr:nth-child(odd) td{background-color:#FFFFFF;}
#Result_Table table tr:nth-child(even) td{background-color:#E6E6FA;}
#Result_Table{max-width:80%;margin-left:18%;}

#Result_Novel td {font-size:14px;border-collapse:collapse;text-align:center;}
#Result_Novel th{background-color:#8B0000;color:#FFFFFF;border-color:#FFFFFF;text-
align:center;}
#Result_Novel table tr:nth-child(odd) td{color:#8B0000;background-color:#FFFFFF;}
#Result_Novel table tr:nth-child(even) td{color:#8B0000;background-color:#FFB2B2;}
#Result_Novel {max-width:80%;margin-left:18%;}

#footer{color:#FFFFFF;font-size:10pt; padding: 10px 10px; background-color:#003366;border-radius:
4px;}
</style>

```

```

<title>tbvar</title>
</head>

```

```

<body style="font-family: Arial;">
<div id=Header style="vertical-align:middle; height:100;" >
<a href="/Index.html" title="Go to tbvar Home Page" ></a></div>
<br>
<div id=Links><b>
<a href="/Index.html" ></a>
<a href="/annoTB.html" ></a>
<a href="/help.html" ></a>
<a href="/about.html"></a>
<a href="/contact.html"></a>
</b>
</div>;

```

```

#####Connecting to the database#####
my $dbh=DBI->connect('dbi:mysql:mtb_anno','root','123456') or die "Connection Error:$DBI::errstr\n";

my $sql1="Select * from mtb_anno.tbvar_db";
my $sql2="Select * from mtb_anno.drug";

my @pos=split(/,,$query); my $uploaded=@pos;
my $sql1="Select * from mtb_anno.tbvar_db where var_int_id=$pos[0]";
my $sql2="Select * from mtb_anno.drug where var_int_id=$pos[0]";

for(my $i=1;$i<$uploaded;$i++)

```



```

print"</tr></table></div>";
print"<br><p style='margin-left:150px'><b>Synonymous Variations</b><br>
<div id=Result_Table><table><tr><th>Variant Position</th><th>Ref Allele</th><th>Alt
Allele</th><th>Variant Count</th><th>Gene</th><th>Type</th></tr><tr>";
$sth1->execute or die "SQL error:$DBI::errstr\n";
while(my @all_1= $sth1->fetchrow_array)
{
my $freq_pct=sprintf "%.2f",(($all_1[6]/470)*100);
if($all_1[5] eq 'synonymous SNV')
{print"<td>$all_1[0]</td><td>$all_1[2]</td><td>$all_1[3]</td><td>$freq_pct</td><td><a
href='http://tuberculist.epfl.ch/quicksearch.php?gene+name=$all_1[8]&submit=Search'>$all_1[8]</a></td><td>
>$all_1[5]</td></tr><tr>";
}
}

```

```

print"</tr></table></div>";
print"<br><p style='margin-left:150px'><b>Regulatory Variations</b><br>
<div id=Result_Table><table><tr><th>Variant Position</th><th>Ref Allele</th><th>Alt
Allele</th><th>Variant
Count</th><th>Gene</th><th>Type</th><th>Regulator</th><th>Target</th></tr><tr>";
$sth1->execute or die "SQL error:$DBI::errstr\n";
while(my @all_1= $sth1->fetchrow_array)
{
if($all_1[15])
{
my $freq_pct=sprintf "%.2f",(($all_1[6]/470)*100);
    $all_1[15]=substr($all_1[15],1);
    @reg=split(/;/,$all_1[15]);
    $all_1[16]=substr($all_1[16],1);
    @targets=split(/;/,$all_1[16]);
    $star=@reg;
    for(my $i=0;$i<$star;$i++)
    {print"<td>$all_1[0]</td><td>$all_1[2]</td><td>$all_1[3]</td><td>$freq_pct</td><td><a
href='http://tuberculist.epfl.ch/quicksearch.php?gene+name=$all_1[8]&submit=Search'>$all_1[8]</a></td><td>
>$all_1[5]</td><td>$reg[$i]</td><td>$targets[$i]</td></tr><tr>";}
}
}

```

```
print"</tr></table></div>";
```

```

print"<br><p style='margin-left:150px; color:#8B0000'><b>Novel Variations</b><table><td></td><td>
align='right'><form action='/cgi-bin/annoTB_submit.cgi' method='post'><input name='hidden' type='hidden'
value='@text'><input type='submit' name='submit' value='Submit'></form></td></table>
<div id=Result_Novel>
<table><tr><th>Variant Position</th><th>Ref Allele</th><th>Alt Allele</th></tr><tr>";

```

```
my $flag;my @check;
```

```

foreach my $p(@pos)
{ $flag=0;
foreach my $m(@mapped)
{
if($p==$m)
{ $flag=1; next;}
}
push(@check,$flag);
}
my $c=@check;

```

```
for(my $i=0;$i<$c;$i++)
{
if($check[$i]==0)
{
print"<td>$pos[$i]</td><td>$ref[$i]</td><td>$alt[$i]</td></tr><tr>";
}}

print"</tr></table><br><center></center></div><br><br><br>
<div id=footer style='height:30px;margin-right:5px;width=100; color:#FFFFFF;font-size:10pt; padding: 10px
10px; background-color:#003366;'><b>Copyright&copy;2013 CSIR";

#####
```

APPENDIX-VI

CGI script to save the novel variations submitted by the user on the server:

```
#!/usr/bin/perl

use CGI;

##### Fetching the submitted data#####
my $cgi = new CGI;
my $text=$cgi->param('hidden');
    my $name=$cgi->param('name');
    my $insti=$cgi->param('insti');
    my $email_address = $cgi->param('id');
    my $strain=$cgi->param('strain');
    my $pub=$cgi->param('publication');
    my $ref_genome = $cgi->param('reference');
    my $location = $cgi->param('location');
    my $depth = $cgi->param('coverage');

my @split_id=split(/@/, $email_address);
my $file_name=$strain.'_'. $split_id[0].'.txt';

my @input=split(/\s/, $text);

#####Designing the web interface#####
print "Content-type:text/html\r\n\r\n";

print '<html><head>
<title>tbvar</title>
</head>
<body style="font-family: Arial;">

<div id=Header style="vertical-align:middle;"><table id="headertab" width="100%" bgcolor="#003366">
<tr><td><a href="/Index.html" title="Go to tbvar Home Page" ></a></td><!--td
align="right"><form style="display:inline" id="submit" action="tbvar.cgi" method="Get"><input type="text"
id="query" name="query" size="30" placeholder="Search: Variant Location, Range or Rvid"><input
name="hidden" type="hidden" value="variation"><input type="submit"
value="Eureka"></center></form></td--><td align="right"></td></tr></table></div>
<br>

<div id=Links><b>
<a href="/Index.html" ></a>
<a href="/annoTB.html" ></a>
<a href="/help.html" ></a>
<a href="/about.html"></a>
<a href="/contact.html"></a>
</b>
</div><br><br>';

#####To accept the submission onlyif the complete form has been filled#####
if(!$name && !$insti && !$email_address && !$strain && !$pub && !$ref_genome && !$location &&
!$depth )
{
print"
<center>
```

```

<div id=Submission_form style='text-align: justify; margin-left:100px;margin-right:50px;'><form action='/cgi-
bin/annoTB_submit.cgi' method='post'>
Your Name:<br><input type='text' id='name' name='name' size='30'><br><br>
Institution:<br><input type='text' id='insti' name='insti' size='30'><br><br>
e-mail ID:<br><input type='text' id='id' name='id' size='30'><br><br>
Strain: <br><input type='text' id='strain' name='strain' size='30'><br><br>
Publication: <br><input type='text' id='publication' name='publication' size='30'><br><br>
Reference Genome: <br><input type='text' id='reference' name='reference' size='30'><br><br>
Geographic Location: <br><input type='text' id='location' name='location' size='30'><br><br>
Minimum Depth Coverage: <br><input type='text' id='coverage' name='coverage' size='30'><br><br>
<input name='hidden' type='hidden' value='@input'>
<br><br><input type='submit' name='send' value='Submit'></form>
</div></center>";
}
else
{

print<center><p style="color:#003366;font-size: 28px ;">Thanks for submission.</p></center></body>;
open(info,">upload/$file_name");
print info "## $name\n## $insti\n## $email_address\n## $strain\n## $pub\n## $ref_genome\n## $location\n##
$depth\n\n";
foreach (@input)
{
print info "$_\n";
}
}

print"<div id=footer style='height:30px;margin-right:5px;width=100; color:#FFFFFF;font-size:10pt; padding:
10px 10px; background-color:#003366;'><b>Copyright&copy;2013 CSIR</b></div>";

```

APPENDIX-VII

CGI script to retrieve the information fed by the user in the contact page and save it on the server:

```
#!/usr/bin/perl

use CGI;
#####Extracting the submitted information#####
my $cgi = new CGI;
    my $name=$cgi->param('name');
    my $email_address = $cgi->param('id');
    my $feedback = $cgi->param('feedback');

my @split_id=split(/@/, $email_address);
my $file_name=$split_id[0].'.txt'; ##### Creating the file name in which the feedback info
#####would be saved

#####Designing the web page#####
print "Content-type:text/html\r\n\r\n";
print '<html><head>
<title>tbvar</title>
</head>
<body style="font-family: Arial;">
<div id=Header style="vertical-align:middle;"><table id="headertab" width="100%" bgcolor="#003366">
<tr><td><a href="/Index.html" title="Go to tbvar Home Page" ></a></td><!--td
align="right"><form style="display:inline" id="submit" action="tbvar.cgi" method="Get"><input type="text"
id="query" name="query" size="30" placeholder="Search: Variant Location, Range or Rvid"><input
name="hidden" type="hidden" value="variation"><input type="submit"
value="Eureka"></center></form></td--><td align="right"></td></tr></table></div>
<br>
<div id=Links><b>
<a href="/Index.html" ></a>
<a href="/annoTB.html" ></a>
<a href="/help.html" ></a>
<a href="/about.html"></a>
<a href="/contact.html"></a>
</b>
</div><br><br>';

#####Accepting the submission only if the complete form has been filled#####
if(!$name && !$email_address && !$feedback)
{
print'
<div id=Feedback_form style="text-align: justify; margin-left:100px;margin-right:50px;"><form action="/cgi-
bin/feedback_form.cgi" method="post">
Your Name:<br><input type="text" id="name" name="name" size="30"><br><br>e-mail ID:<br><input
type="text" id="id" name="id" size="30"><br><br>Your Comments:<br><textarea name="feedback"
cols="50" rows="10"></textarea>
<br><br><input type="submit" name="send" value="Submit"></form>
</div>
';
}
```

